



**HAL**  
open science

## Discovering of the unobservable behaviour of an Interpreted Petri Net model

Francesco Basile, Gregory Faraut, Luigi Ferrara, Jean-Jaques Lesage

► **To cite this version:**

Francesco Basile, Gregory Faraut, Luigi Ferrara, Jean-Jaques Lesage. Discovering of the unobservable behaviour of an Interpreted Petri Net model. 58th IEEE Conf. on Decision and Control (CDC'19), Dec 2019, Nice, France. pp.2021-2026. hal-02415453

**HAL Id: hal-02415453**

**<https://hal.science/hal-02415453>**

Submitted on 17 Dec 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Discovering of the unobservable behaviour of an Interpreted Petri Net model

Francesco Basile<sup>1</sup>, Gregory Faraut<sup>2</sup>, Luigi Ferrara<sup>1</sup> and Jean-Jaques Lesage<sup>2</sup>

**Abstract**—This paper focuses on the problem of discovering a Petri Net model from long event sequences generated by a discrete event system. Precisely, it is assumed that the relations between input and output events (*i.e.* the observable behaviour of the system) are already modelled by a set of Interpreted Petri Net fragments while the behaviour of the internal state evolutions (*i.e.* the unobservable behaviour) must be discovered. An approach inspired to net synthesis is proposed. It relies on an optimization-based procedure for the identification of the unobservable net structure and marking.

## I. INTRODUCTION

The methods presented in the literature for the identification of DESs produce a mathematical model expressed as a Petri Net (PN) or a finite state automaton model of the system behaviour from sequences observed during the system operation [5]. When the resulting model is a PN, like in this paper, the net structure (places, transitions and arcs) and its initial marking must be computed. The language of the identified model, that is the set of sequences it can generate, in general contains a subset of sequences that do not belong to the observed language. Such a subset represents the exceeding language of the identified system. The size of the exceeding language is a measure of the fitness of the obtained model. Indeed, a large exceeding language is certainly undesired when the identified model is used for diagnostic or verification purposes.

In this work, the identification of closed-loop controlled automation systems is considered. The behaviour of these systems can be split into an *Observable behaviour*, related to direct output changes depending on input changes, and an *Unobservable behaviour*, related to evolutions of the internal state (and variables) of the system without changes of observable data (inputs and outputs). An identification algorithm should provide a model expressing both I/O causal relationships and internal state evolutions due to input changes [15]. This paper can be positioned as the continuation of [10], as the papers [11], [12], [15]. The authors of [10] provide an approach to discover the observable behaviour as Interpreted PN (IPN) fragments from an I/O observed sequence. This sequence is also converted into a firing sequence on the alphabet of transitions (see first step in Fig. 1). Therefore, the unobservable behaviour is discovered from such a firing sequence, and the IPN fragments are completed by adding connecting (unobservable) places (see second step in Fig. 1); these places implement a proper ordering of the transition firings and, thus, are essential in reducing the size of the exceeding behaviour.

Solutions to implement the second step have been presented in [11], [12], [15]. In [11], [12] it is implemented

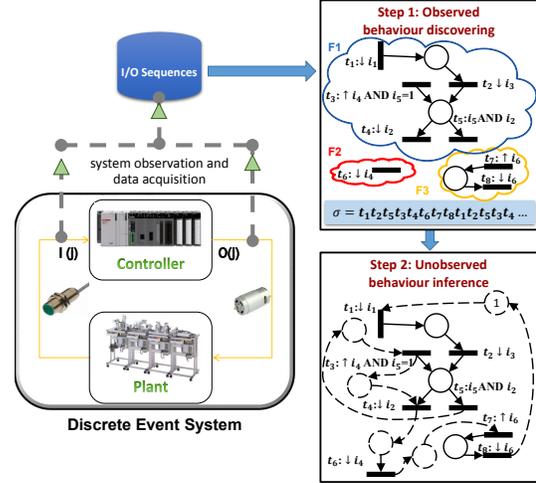


Fig. 1. Identification procedure of closed-loop systems in two steps.

discovering the causal and concurrent relationships between transition firings in the firing sequence obtained in the first step. In [15] the projection of the firing sequence obtained in the first step on subalphabets is used to discover specific patterns that are characteristic of dependency relationships between the transition firings. Both the approaches return as identified model a 1-bounded net.

In this paper a different procedure is used to discover the unobservable behaviour.

First, a synthesis approach is used to solve the problem of reducing the exceeding language. There are approaches to DES identification where it is assumed that either the whole state space of the system, or the whole language generated by it, is known [6], [7], [8]. If this is the case, the tackled problem is more a *net synthesis* problem, rather than a *net identification* one. In this paper a net synthesis approach based on a graph is used. Precisely, the approach presented in this paper, inspired to [13], forces the reachability graph of the identified model to be isomorphic to a graph generating a behaviour having empty exceeding language with respect to words of length  $r$ , where  $r$  is a design parameter. Moreover, the observable net, as well as the identified unobservable one, are not required to be 1-bounded.

Second, the sequences of transition firings of the observable net obtained as result of the first stage of the procedure depicted in Fig. 1, are enriched by the observable markings reached during the firing of the sequence of transitions by the observable net model. The goal is the definition of an exceeding language with respect to sequences of transitions (associated to system inputs) and markings (associated to system outputs) and not transition only to improve the quality of the identified model in terms of accuracy. Indeed,

<sup>1</sup> F. Basile and L. Ferrara are with DIEM, Università di Salerno, Italy.

<sup>2</sup> G. Faraut and J.J. Lesage are with LURPA, ENS Cachan, Univ. Paris-Sud, Univ. Paris-Saclay, France.

assume that a sequence  $t_1 t_2$  is generated by an identified net model; if we consider also system outputs, it may happen that  $m_1 t_1 m_2 t_2 m_3$  and  $m_4 t_1 m_5 t_2 m_6$  are generated by the identified model, while only  $m_1 t_1 m_2 t_2 m_3$  has been observed.

Third, an optimization approach based on an Integer Linear Programming (ILP) formulation is used for the synthesis of unobservable places. This is in line with a recent trend in PN-related research (see, e.g., [17], [9]).

## II. PN BACKGROUND

A brief recall on Petri Nets is presented in this section. For a complete review on PNs, the reader can refer to [14]. In the following, given a set  $A$ , the cardinality of  $A$  is denoted by  $|A|$ .

A *Place/Transition net* (P/T net) is a 4-tuple  $N = (P, T, \mathbf{Pre}, \mathbf{Post})$  where:  $P$  is a set of places,  $T$  is a set of transitions and  $\mathbf{Pre}$  and  $\mathbf{Post}$  are the  $|P| \times |T|$  sized, natural valued, incidence matrices;  $\mathbf{Post}(p, t) = w$  means that there is an arc from  $t \in T$  to  $p \in P$  with weight  $w$ . A *marking* (the net state) is a vector  $\mathbf{m} : P \rightarrow \mathbb{N}$  that assigns to each place a nonnegative integer number of tokens. The incidence matrix  $C$  of the net is  $C = \mathbf{Post} - \mathbf{Pre}$ . A *P/T system* or *net system*  $\langle N, \mathbf{m}_0 \rangle$  is a net  $N$  with an initial marking  $\mathbf{m}_0$ . A transition  $t$  is *state-enabled* at  $\mathbf{m}$  iff  $\mathbf{m} \geq \mathbf{Pre}(\cdot, t)$ ; its firing yields a new marking  $\mathbf{m}' = \mathbf{m} + C(\cdot, t)$ .

To make explicit the interaction of a PN model with the external environment, we make use of Interpreted Petri Nets (IPNs), whose structure is defined as  $N = (\bar{P}, T, \bar{\mathbf{Pre}}, \bar{\mathbf{Post}}, \gamma, \beta)$ . The first advantage of IPNs is the possibility to characterize each transition with a logic condition depending on the system's inputs; in particular, the function  $\beta : T \rightarrow \{0, 1\}$  associates a logical condition to transitions  $\forall t_i \in T, \beta(t_i) = F_i(I, E_I)$ , where  $I$  is the set of input signals and  $E_I = \{\uparrow i_j (\downarrow i_j) \mid i_j \in I\}$  is the set their rising (falling) edges. Secondly, the system's outputs are explicitly represented in the net; in particular, the set of places  $\bar{P}$  is partitioned into observable  $P$  and unobservable  $P^u$  ( $\bar{P} = P \cup P^u$ ,  $P \cap P^u = \emptyset$ ) and, denoted by  $O$  the set of output signals, the function  $\gamma : \bar{P} \rightarrow O \cup \{\epsilon\}$  associates to each observable place an output signal and to each unobservable place the symbol  $\epsilon$  denoting a null output. Thus, observable places are used to model the output configurations of the DES while the unobservable ones implement a proper ordering of the transition firings.

An *IPN system*  $\langle N, \bar{\mathbf{m}}_0 \rangle$  is an IPN  $N$  with an initial marking  $\bar{\mathbf{m}}_0$ ; a transition  $t$  fires at  $\bar{\mathbf{m}}$  if and only if it is state enabled and logical condition enabled ( $\beta(t) = 1$ ).

## III. PROBLEM STATEMENT

We assume that an IPN system  $\langle N, \bar{\mathbf{m}}_0 \rangle$  modelling the system of interest is available. In addition, the set of I/O sequences acquired from the system is available in the form of *interpreted sequences*, i.e. sequences consisting of transition firings and observable markings; such set is denoted by  $I\Sigma_{Obs}$ . The  $j$ -th interpreted sequence in  $I\Sigma_{Obs}$  is defined as  $i\sigma_j = \mathbf{m}_{0,j} t_{1,j} \mathbf{m}_{1,j} \dots t_{l-1,j} \mathbf{m}_{l-1,j}$ , being  $l$  its length; the length of  $i\sigma_j$  is denoted by  $|i\sigma_j|$ . Notice that  $t_{i,j}$  ( $\mathbf{m}_{i,j}$ ) refers to the transition (observable marking) occupying the  $i$ -th position in the  $j$ -th observation, where  $1 \leq j \leq |I\Sigma_{Obs}|$ . Each  $i\sigma_j \in I\Sigma_{Obs}$  is supposed to represent a different trajectory of the system starting from the same initial state; as a consequence  $\mathbf{m}_{0,j} = \mathbf{m}_0, \forall i\sigma_j \in I\Sigma_{Obs}$ .

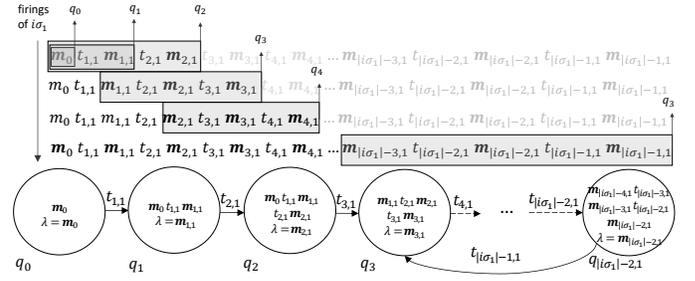


Fig. 2. Creation of the Moore Machine  $E_r$  for  $r = 4$ .

The minimal requirement for  $\langle N, \bar{\mathbf{m}}_0 \rangle$  is to *simulate* the observations: called  $I\Sigma(N, \bar{\mathbf{m}}_0)$  the set of interpreted sequences of any length that  $N$  can generate from  $\bar{\mathbf{m}}_0$ , it must hold  $I\Sigma_{Obs} \subseteq I\Sigma(N, \bar{\mathbf{m}}_0)$ .

Given an interpreted sequence  $i\sigma_j \in I\Sigma(N, \bar{\mathbf{m}}_0)$ , the subsequence of length  $n \leq |i\sigma_j|$  starting from the  $k$ -th marking is a *word*  $w_k^n(i\sigma_j)$ , where  $w_k^n \equiv w_k^n(i\sigma_j) = \mathbf{m}_{k,j} t_{k+1,j} \mathbf{m}_{k+1,j} \dots \mathbf{m}_{k+n-2,j} t_{k+n-1,j} \mathbf{m}_{k+n-1,j}$ ,  $0 \leq k \leq |i\sigma_j| - n$ . Words  $w_k^n$  associated to  $i\sigma_j \in I\Sigma_{Obs}$ , are called *known* since they directly originate from the observations. Given a length  $n$ , they can be collected in the set of known words of length  $n$ ,  $K^n = \{w_k^n, 0 \leq k \leq |i\sigma_j| - n, i\sigma_j \in I\Sigma_{Obs}\}$ , while  $L_{Obs}^n = \{w_{0,j}^l, l \leq n, i\sigma_j \in I\Sigma_{Obs}\}$  is called *observed language* of length  $n$  and collects known words of maximum length  $n$  starting from the initial marking. Similarly, the set of *unknown words* of length  $n$  is defined  $U^n(N, \bar{\mathbf{m}}_0) = \{w_k^n, \forall 0 \leq k \leq |i\sigma_j| - n, i\sigma_j \in I\Sigma(N, \bar{\mathbf{m}}_0) \wedge w_k^n \notin K^n\}$ , and is due to unexpected transition firings.

Simulation admits that the model can enable unexpected transition firings at each marking in addition to expected ones. On the contrary, an accurate model only enables expected transitions; formally, denoted by  $L^n(N, \bar{\mathbf{m}}_0) = \{w_{0,j}^l, l \leq n, i\sigma_j \in I\Sigma(N, \bar{\mathbf{m}}_0)\}$  the language of length  $n$  generated by the identified net system from  $\bar{\mathbf{m}}_0$ , the *exceeding language*  $L_{Exc}^n(N, \bar{\mathbf{m}}_0) = L^n(N, \bar{\mathbf{m}}_0) \setminus L_{Obs}^n$  is empty in an accurate model for each length  $n$ . Maximum accuracy, however, is quite never the target for a good model. In fact, since real systems usually exhibit a rich behaviour, long sequences are needed to capture them; however, observations are generally not complete in practice, i.e. not all the possible trajectories are observed. Thus, building a model with maximum accuracy implies that new (future) observations cannot be generated by the identified model.

The empirical experience usually suggests that it makes sense to devise a tolerance length  $\tilde{n}$  able to preserve a rich set of observations. In this case a good model is such that, not only  $L^{\tilde{n}}(N, \bar{\mathbf{m}}_0) = \emptyset$  holds, but also every produced word of length  $n \leq \tilde{n}$  is a known word; more formally, called *distance* of length  $n$  the quantity  $d^n(N, \bar{\mathbf{m}}_0) = |U^n(N, \bar{\mathbf{m}}_0) \setminus K^n|$ , the second constraint is expressed by  $d^{\tilde{n}}(N, \bar{\mathbf{m}}_0) = 0$ . Obviously  $\tilde{n}$  cannot exceed the length of the longest observation.

The goal of this paper is to refine an IPN model to make it more accurate. At this aim it is assumed that:

- 1) the automation system is already modelled by an IPN system  $\langle N^{Obs}, \mathbf{m}_0 \rangle$  and the set  $I\Sigma_{Obs}$  is available; it is also called *observable net*, since it represents the I/O observable behaviour of the automation system and only

contains observable places and transitions [12]; obviously,  $\langle N^{Obs}, \mathbf{m}_0 \rangle$  must simulate the acquired observations;

- 2) the refinement process consists in identifying the marked unobservable places to be added to  $\langle N^{Obs}, \mathbf{m}_0 \rangle$ ;
- 3)  $\tilde{n}$  is given.

It is useful to characterize the accuracy through the design parameter  $r \geq \tilde{n}$ . The objective is thus to construct a new IPN system  $\langle N', \bar{\mathbf{m}}_0 \rangle$  such that: 1)  $d^r(N', \bar{\mathbf{m}}_0) = 0$ ; 2)  $L_{Exc}^r(N', \bar{\mathbf{m}}_0) = \emptyset$ ; If 1) and 2) hold, the new IPN system is said to be  $r$ -complete. The two constraints are both necessary: if only the first one is imposed, the model just produces known words of length  $n \leq r$ , but some words could be produced from the initial marking even if they were observed from a state of the system different from the initial one; if the second one only is imposed, unknown words of any length  $n$  could be produced after the first  $r - 1$  firings from the initial marking. Note that  $r$ -completeness entails  $\tilde{n}$ -completeness ( $r \geq \tilde{n}$ ).

#### IV. A SOLUTION BASED ON ILP PROBLEMS SOLVING

A net synthesis approach is proposed to solve the identification problem stated in the previous section.

This task is accomplished by firstly constructing a Moore machine  $E_r$  representing the dynamics that  $\langle N', \bar{\mathbf{m}}_0 \rangle$  should exhibit. For the sake of simplicity, in this paper we suppose that if it holds  $w_{0,i}^{r-1} = w_{k,j}^{r-1}$ ,  $k > 0$ ,  $i\sigma_i, i\sigma_j \in I\Sigma_{Obs}$  then it also holds  $w_{0,i}^r = w_{k,j}^r$ ; therefore, if a word of length  $r - 1$  is produced both at the beginning of any observation and also later, it must be followed by the same transition firing and observable marking. This assumption can be easily removed by slightly modifying the algorithm illustrated in the following.

The algorithm is based on a sliding window mechanism. This window has a dynamic size and slides on each  $i\sigma \in I\Sigma_{Obs}$ ; at any moment, it only contains a sub-word of  $i\sigma$ . The automaton is constructed by adding at each step a state  $q$  associated to the sub-word  $w$  contained in the window, briefly denoted by  $w = \omega(q)$ ; in addition, to each state  $q$  an observable marking  $\mathbf{m}$  is associated as label, briefly denoted by  $\mathbf{m} = \lambda(q)$ . We also denote by  $last(w)$  the last marking of the word  $w$ . In detail:

- 1) The interpreted sequence  $i\sigma_j, j = 1$  is firstly selected. The window is positioned at the beginning of  $i\sigma_1$  ( $k = 0$ ); the length of the window is  $l = 1$  and contains the word  $w_0 = w_{k,j}^l = w_{0,1}^1 = \mathbf{m}_0$ : the initial state  $q_0$  is created, such that  $\omega(q_0) = w_0$ ,  $\lambda(q_0) = last(w_0) = \mathbf{m}_0$ . The window's length  $l$  is then incremented by a unit and the word  $w_1 = w_{0,1}^2$  is encountered. A state  $q_1$  is thus constructed, such that  $\omega(q_1) = w_1$ ,  $\lambda(q_1) = last(w_1) = \mathbf{m}_{1,1}$ ; the state  $q_0$  is linked to  $q_1$  by an arc labelled by  $t_{1,1}$ . This procedure repeats until  $l = r - 2$ ; then,  $l = r - 1$  is set (maximum length of the window) and the word  $w = w_{0,1}^{r-1}$  is encountered. The state  $q_{r-3}$  is linked to  $q_{r-2}$  through an arc labelled by  $t_{r-2,1}$ ;
- 2) The window, now, slides by one step ( $k = 1$ ) keeping the same length; the contained word is  $w' = w_{1,1}^{r-1}$ . A new state  $q_{r-1}$  is created and associated to  $w'$ . The previous state is linked to  $q_{r-1}$  by an arc labelled by  $t_{r-1,1}$ . The window then slides again ( $k = 2$ ). This process, whose details are given in the following, continues until the window reaches the end of  $i\sigma_1$ ;
- 3) When the processing for  $i\sigma_1$  is finished, the following interpreted sequence (if it exists) is selected, thus  $j$  is incremented and the procedure re-executed from the step 1); new states and arcs are thus added to  $E_r$ .

Figure 2 shows how steps 1 and 2 are applied on  $i\sigma_1$  when building  $E_r$  for  $r = 4$ .

If, during the process, it happens that a word  $w$  is encountered again in the window, the state associated to  $w$  and created the first time that  $w$  was encountered, is *re-used* and new input/output arcs added to it. In Fig. 2 it is exemplified the case where the last word encountered in the window, *i.e.*  $w = \mathbf{m}_{|i\sigma_1|-3,1}t_{|i\sigma_1|-2,1}\mathbf{m}_{|i\sigma_1|-2,1}t_{|i\sigma_1|-1,1}\mathbf{m}_{|i\sigma_1|-1,1}$  is equal to the word already associated to  $q_3$ , *i.e.*  $w = \mathbf{m}_{1,1}t_{2,1}\mathbf{m}_{2,1}t_{3,1}\mathbf{m}_{3,1}$ ; thus  $q_3$  is re-used and  $t_{|i\sigma_1|-1,1}$  is added as input arc coming from the state associated to  $w = \mathbf{m}_{|i\sigma_1|-4,1}t_{|i\sigma_1|-3,1}\mathbf{m}_{|i\sigma_1|-3,1}t_{|i\sigma_1|-2,1}\mathbf{m}_{|i\sigma_1|-2,1}$ .

As a result of state re-using, a single state exists for each encountered word in the window. At the end of the procedure, the set of words associated to the states of  $E_r$  is  $\Phi(r) = \{L_{Obs}^{r-2} \cup K^{r-1}\}$ , since states associated to words in  $L_{Obs}^{r-2}$  are the ones created at step 1, while states associated to words in  $K^{r-1}$  are the ones created at step 2.

Once  $E_r$  is built, a word  $w$  can be associated to each direct path belonging to  $E_r$ . Precisely, given a direct path from a state  $q_i$  to a state  $q_j$ , the word  $w = \lambda(q_i)t_i\lambda(q_{i+1})t_{i+1}\dots t_{j-1}\lambda(q_j)$  can be devised, where  $t_k$  is one of the transitions associated to the arcs exiting from  $q_k$ . Once these words have been defined, as done for IPNs, language and distance can be introduced in a similar way also for  $E_r$ . The automaton  $E_r$  is  $r$ -complete by construction.

Being the net system  $\langle N^{Obs}, \mathbf{m}_0 \rangle$  not  $r$ -complete, some sequences exist that can be produced in  $N^{Obs}$  from  $\mathbf{m}_0$  but not in  $E_r$  from  $q_0$ . In the general case, each one of these sequences can be split in two parts: the first subsequence (that can be empty) is still feasible in  $E_r$ ; the second one begins with the first infeasible transition firing, *i.e.* the first transition that is not expected in  $E_r$ . The general idea is to add in the IPN system a new unobservable place that blocks the first unexpected transition firing of the infeasible (sub)path. These places are synthesized by solving ILP problems based on proper algebraic constraints, which are split into *global* and *local* constraints.

#### A. Global algebraic constraints

In this section, we devise the set of constraints that each unobservable place of the new IPN system  $\langle N', \bar{\mathbf{m}}_0 \rangle$  must ensure; more specifically:

- (a) each interpreted sequence in  $I\Sigma_{Obs}$  must fire from the initial marking;
- (b) each time a word  $w \in \Phi(r)$  is generated by the IPN, the marking reached at the end of  $w$  must be always the same.

In order to formalize these constraints, we firstly denote the sequence of transitions associated to  $i\sigma_j$  by  $\sigma_j = t_{1,j}t_{2,j}\dots t_{|i\sigma_j|-1,j}$ ; denoted by  $|\sigma_j|$  its length, it holds  $|i\sigma_j| = |\sigma_j| + 1$ . Furthermore, the following definitions come in help:

- 1) The function  $\Theta : w \rightarrow \{\sigma\}$ , defined for each  $w \in K^n$ , yields the observed (sub)sequences of transitions starting from the initial marking that produce  $w$  in the last  $|w| - 1$  firings, *i.e.*  $\Theta(w) = \{\sigma_j(1, k + |w| - 1) \mid \exists w_{k,j}^{|w|} = w, i\sigma_j \in I\Sigma_{Obs}\}$ , where  $\sigma(m, n)$  is the subsequence of  $\sigma$  having all its elements from the  $m$ -th to the  $n$ -th; if  $n < m$  the empty sequence is returned.
- 2) Consider an IPN  $N$  and let  $p^U$  be an unobservable place of  $N$ ;  $p^U$  is said to be  $r$ -consistent if and only if  $\forall w \in \Phi(r)$  it holds  $\mathbf{c}^U \sigma_i = \mathbf{c}^U \sigma_j$  where  $\mathbf{c}^U$  is the row of

the incidence matrix of  $N$  associated to  $p^U$ ,  $\sigma_i, \sigma_j \in \Theta(w)$ ,  $i \neq j$  and  $\sigma \in \mathbb{N}^{|\mathcal{T}|}$  is the firing count vector associated to  $\sigma$  whose  $k$ -th component is the number of occurrences of  $t_k$  in  $\sigma$ . If  $\sigma = t_i$ , then the associated firing count vector is denoted by  $\mathbf{t}_i$ .

The previously enumerated constraints can now be formally expressed. Chosen  $r$ , any unobservable place  $p^U$  of  $\langle N', \bar{\mathbf{m}}_0 \rangle$ , identified by the incidence rows  $\mathbf{pre}^U$ ,  $\mathbf{post}^U$  and the initial marking  $m_0^U$ , must ensure the following algebraic system:

$$\begin{cases} m_0^U + \mathbf{c}^U \tilde{\sigma}_1 \geq \mathbf{pre}^U \tilde{\sigma}_2, & (1) \\ \tilde{\sigma}_1 = \sigma_k(1, l-1), \tilde{\sigma}_2 = \sigma_k(l), \\ \forall i \sigma_k \in I\Sigma_{Obs}, \forall 1 \leq l \leq |\sigma_k|; \\ \mathbf{c}^U \sigma_i = \mathbf{c}^U \sigma_j, & (2) \\ \forall \sigma_i, \sigma_j \in \Theta(w), \forall w \in \Phi(r), i \neq j; \end{cases}$$

where  $\mathbf{c}^U = \mathbf{post}^U - \mathbf{pre}^U$ , since global constraint (a) is implemented by inequalities (1) that admit legal firings, and global constraint (b) is implemented by equations (2) that ensure  $r$ -consistency.

### B. Local algebraic constraints

In this section, the *sets* of *local* constraints are presented; we use the term *local* to remark that each *set* will be implemented by a different unobservable place, which also implements global constraints.

Local constraints are responsible for enforcing the disabling of undesired transition firings; as a consequence, they depend on the accuracy of the given net. For sake of generality, it is convenient to suppose that the given net system  $\langle N, \bar{\mathbf{m}}_0 \rangle$  satisfies the hypothesis and already contains some  $r$ -consistent unobservable places.

Let us consider the word  $w \in \Phi(r)$  associated to a state of  $E_r$ . Constraint (2) guarantees that, by firing from the initial marking any sequence  $\sigma$  that produces  $w$  in the last  $|w| - 1$  firings ( $\sigma \in \Theta(w)$ ), the unobservable places always reach the same marking. The same already occurs for the observable places, by definition of interpreted sequence that always terminates with an observable marking. Consequently, the set of state-enabled transitions by the IPN system after the generation of  $w$  is uniquely determined by  $w$  and we denoted it by  $\mathcal{E}(N, w)$ . In addition we denote by  $\mathcal{A}_r(w)$  the set of transitions associated to the output arcs of the state of  $E_r$  associated to  $w$ .

Now, local constraints can be formally expressed as follows. Chosen a word  $w \in \Phi(r)$  and selected a  $\sigma_i \in \Theta(w)$ , the unobservable place  $p^U$  identified by the incidence rows  $\mathbf{pre}^U$ ,  $\mathbf{post}^U$  and the initial marking  $m_0^U$ , disables the undesired firing of a transition  $t_u \in \{\mathcal{E}(N, w) \setminus \mathcal{A}_r(w)\}$  after the firing of  $\sigma_i$  from  $m_0^U$  iff:

$$\begin{cases} m_0^U + \mathbf{c}^U \sigma_i < \mathbf{pre}^U \mathbf{t}_u; & (3) \\ \mathbf{pre}^U \mathbf{t}_u > 0; & (4) \end{cases}$$

where  $\mathbf{c}^U = \mathbf{post}^U - \mathbf{pre}^U$ , since by inequality (3) the disabling of the undesired firing is accomplished while by (4) the blocking arc is imposed to the transition  $t_u$ .

### C. The core algorithm

In this section, the core algorithm that produces the desired net system  $\langle N', \bar{\mathbf{m}}_0' \rangle$  is presented.

---

#### Algorithm 1: $r$ -completeness enforcement

---

**input** :  $E_r, \Phi(r), I\Sigma_{Obs}, r, \langle N^{Obs}, \mathbf{m}_0 \rangle$ .  
**output**: A new IPN system  
 $\langle N', \bar{\mathbf{m}}_0' \rangle \equiv \langle (\bar{P}', T, \bar{\mathbf{Pre}}, \bar{\mathbf{Post}}, \gamma', \beta'), \bar{\mathbf{m}}_0' \rangle$   
that tries to ensure  $r$ -completeness.

- 1 **Initialize**:  $\langle N', \bar{\mathbf{m}}_0' \rangle = \langle N^{Obs}, \mathbf{m}_0 \rangle$   
 $f_{min} = (\mathbf{pre}^U + \mathbf{post}^U) \cdot \mathbf{1} + m_0^U$
- 2 **for each**  $w \in \Phi(r)$  **do**
- 3     Select a random  $\sigma_i \in \Theta(w)$
- 4      $T_u^{treated} = \emptyset$
- 5     **while**  $\{\{\mathcal{E}(N', w) \setminus \mathcal{A}_r(w)\} \setminus T_u^{treated}\} \neq \emptyset$  **do**
- 6         Select a random
- 7          $t_u \in \{\mathcal{E}(N', w) \setminus \mathcal{A}_r(w)\} \setminus T_u^{treated}$
- 8         Add in the set of constraints, named  $S$ , (1), (2), (3), (4) devised from  $w, t_u$  and  $\sigma_i$
- 9         **if**  $\nexists p^U \in \bar{P}'$  s.t.  $S$  is satisfied **then**  
               a new unobservable place  $p^U$  is constructed:  
                $[\mathbf{pre}^U, \mathbf{post}^U, m_0^U, solved] = solveILP(S, f_{min})$   
               **if**  $solved$  **then**  
                    $\bar{P}' = \bar{P}' \cup p^U, \gamma'(p^U) = \epsilon,$   
                    $\bar{\mathbf{Pre}} = \begin{bmatrix} \mathbf{Pre} \\ \mathbf{pre}^U \end{bmatrix},$   
                    $\bar{\mathbf{Post}} = \begin{bmatrix} \mathbf{Post} \\ \mathbf{post}^U \end{bmatrix}, \bar{\mathbf{m}}_0' = \begin{bmatrix} \bar{\mathbf{m}}_0' \\ m_0^U \end{bmatrix}$
- 10          $T_u^{treated} = T_u^{treated} \cup t_u$

---

**Theorem 1:** The net system  $\langle N', \bar{\mathbf{m}}_0' \rangle$  produced by Algorithm 1 is  $r$ -complete if and only if all the ILP problems admit a solution, otherwise it holds  $d^r(N', \bar{\mathbf{m}}_0') \leq d^r(N^{Obs}, \mathbf{m}_0)$  and  $L_{Exc}^r(N', \bar{\mathbf{m}}_0') \subseteq L_{Exc}^r(N^{Obs}, \mathbf{m}_0)$ .

*Proof:* The algorithm cycles over each word  $w \in \Phi(r)$  and constructs a set of algebraic constraints for each transition  $t_u$  whose firing is not expected after the generation of  $w$ ; if not already implemented by any previously added place, each set is given to an ILP solver which, if it exists, returns as a solution the new unobservable place. Such a place disables the firing of  $t_u$  (due to (3) and (4)) and satisfies the global constraints; thanks to (2) the disabling is performed independently on how  $w$  is generated by the net system, even if enforced on a single  $\sigma_i \in \Theta(w)$ .

If at least one ILP admits a solution, the exceeding language and the distance are reduced since an undesired firing is disabled; if all the sets are implemented, the new system is  $r$ -complete. ■

Note that the unsolvability of some ILPs is due to the nature of a Petri net place: it is mostly like a counter since it only maintains memory of its current status. Being it insensible to the order in which increments and decrements occur, it is not able to implement any constraint that is order-sensible. In addition, note that the algorithm produces unobservable places with minimum arcs weight and minimal

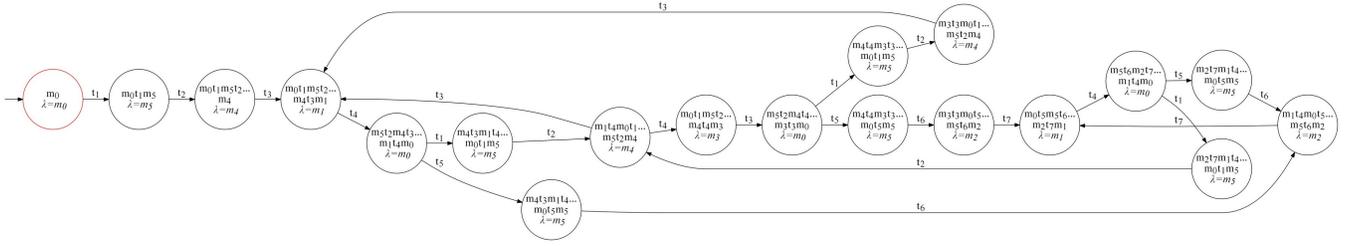


Fig. 3.  $E_r$  for the sorting system and  $r = 5$ .

initial tokens, as the function  $f_{min} = (\mathbf{pre}^U + \mathbf{post}^U) \cdot \mathbf{1} + m_0^U$  imposes, where  $\mathbf{1}$  is a column vector having all elements equal to 1; theorem 1 still holds if the three terms  $(\mathbf{pre}^U, \mathbf{post}^U, m_0^U)$  are differently weighted. The produced net system is in general not 1-bounded.

### V. A CASE STUDY

The purpose of the system in figure 5 is to sort parcels according to their size. It has nine sensors whose signals are  $k_1, k_2, a_0, a_1, a_2, b_0, b_1, c_0, c_1$  and four actuators whose signals are  $A+, A-, B, C$ . The size of a parcel arriving on conveyor 1 is detected as either small ( $k_1$ ) or big ( $k_2$ ). The small (resp. big) parcel is then pushed from the cylinder A to the cylinder B (resp. C) that pushes it on conveyor 2 (resp. 3). The pushing of Cylinder B (C) is detected by means of  $b_0$  ( $c_0$ ) and  $b_1$  ( $c_1$ ). Cylinder A moves the parcel as soon as  $A+$  is asserted; if  $a_1$  (resp.  $a_2$ ) rises, it means that the cylinder B (resp. C) has been reached and, thus, the cylinder A is retracted by enabling  $A-$ , until  $a_0$  rises. The system is sequential: it works only one parcel at a time.

During the observation step described on figure 1, a single observation  $V$  of 222 I/O vectors is carried out, which captures the treatment of 20 parcels and completely acquires the system dynamics. From the analysis of I/O vectors, it results that the treatment of each parcel consists of four transition firings, then it is reasonable to assume  $\tilde{n} = 5$ . Then, we have considered  $r \geq 5$ . Figure 4 shows the first eight I/O vectors in  $V$ ; they correspond to the arrival and sorting of a small parcel.

	$k_1$	$k_2$	$a_0$	$a_1$	$a_2$	$b_0$	$b_1$	$c_0$	$c_1$	$A+$	$A-$	$B$	$C$
$V(1)$	0	0	1	0	0	1	0	1	0	0	0	0	0
$V(2)$	1	0	1	0	0	1	0	1	0	1	0	0	0
$V(3)$	1	0	0	0	0	1	0	1	0	1	0	0	0
$V(4)$	0	0	0	0	0	1	0	1	0	1	0	0	0
$V(5)$	0	0	0	1	0	1	0	1	0	0	1	1	0
$V(6)$	0	0	0	1	0	0	0	1	0	0	1	1	0
$V(7)$	0	0	0	0	0	0	0	1	0	0	1	1	0
$V(8)$	0	0	0	0	0	0	1	1	0	0	1	0	0

Fig. 4. First eight acquired I/O vectors for the sorting system.

The observable IPN system  $\langle N^{Obs}, m_0 \rangle$  is then computed in [16]; it is depicted by solid lines in figure 6. Then  $V$  is translated into an interpreted sequence  $i\sigma = i\sigma_1 i\sigma_2 i\sigma_3 i\sigma_1 i\sigma_3 i\sigma_1 i\sigma_1 i\sigma_3 i\sigma_3 i\sigma_1 i\sigma_3 i\sigma_3 i\sigma_3 i\sigma_3 i\sigma_3 i\sigma_2 i\sigma_1 i\sigma_2 i\sigma_2 i\sigma_1$ , where  $i\sigma_1 = m_0 t_1 m_5 t_2 m_4 t_3 m_1 t_4 m_0$ ,  $i\sigma_2 = m_0 t_1 m_5 t_2 m_4 t_4 m_3 t_3 m_0$  and  $i\sigma_3 = m_0 t_5 m_5 t_6 m_2 t_7 m_1 t_4 m_0$ , where  $m_0 = [0000]^T$ ,  $m_1 = [0010]^T$ ,  $m_2 = [0011]^T$ ,  $m_3 = [0100]^T$ ,  $m_4 =$

$[0110]^T$ ,  $m_5 = [1000]^T$ . The set of observed interpreted sequences is simply  $I\Sigma_{Obs} = \{i\sigma\}$  and the initial marking is  $m_0$ .

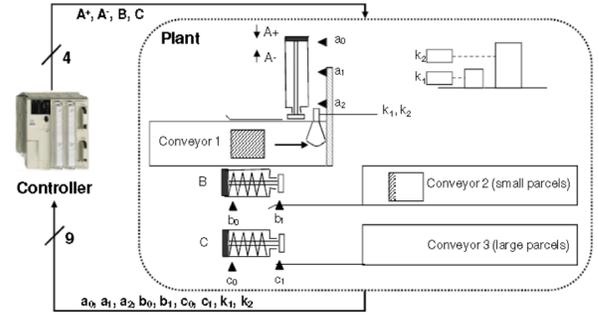


Fig. 5. A simple system to be modelled.

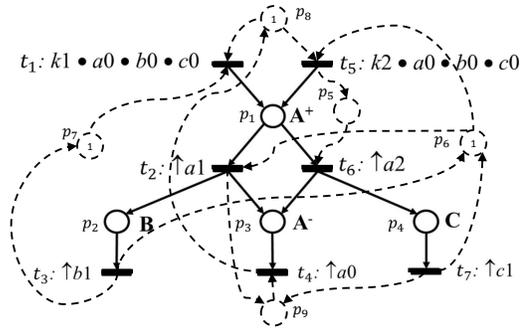


Fig. 6. IPN system  $\langle N', m'_0 \rangle$ .

By choosing  $r = 5$ , the automaton  $E_r$  in figure 3 is obtained. The new Petri net system  $\langle N', \overline{m'_0} \rangle$  is then computed; it is represented in figure 6 (added unobservable places and added arcs in dot lines).

The computed net system is not 5-complete, as figure 7 shows, since  $L_{Exc}^2(N', \overline{m'_0}) \neq \emptyset$ ; for example, the exceeding word  $w = m_0 t_5 m_5$  can be produced from the initial marking, even if it is not expected in  $E_r$ . However, the exceeding language is greatly reduced with respect to the observable IPN system.

In [16] the unobservable places of  $N^{Obs}$  are computed as well, producing the system  $\langle N'', \overline{m''_0} \rangle$ . The comparison with  $\langle N', \overline{m'_0} \rangle$ , shows that both the exceeding language and the distance are larger, as depicted in figure 8. It is interesting to observe that for small values of  $n$  the

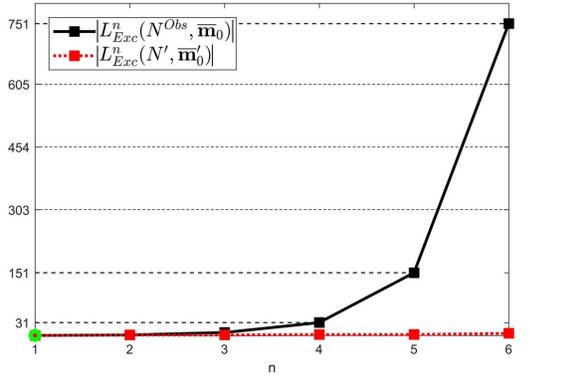


Fig. 7. Comparison between the exceeding language of  $\langle N^{Obs}, \bar{m}_0 \rangle$  and the computed  $\langle N', \bar{m}'_0 \rangle$ . Green points mean no exceeding words.

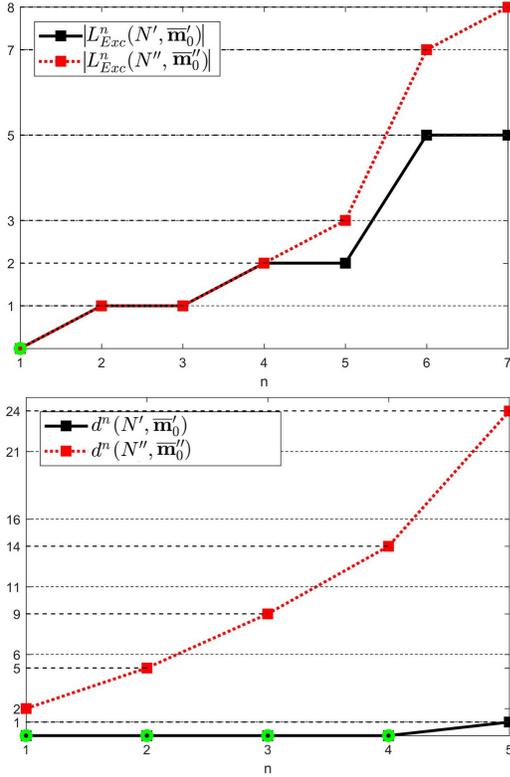


Fig. 8. Comparison of the exceeding language and the distance of the IPN system computed by the proposed method w.r.t the IPN computed in [16].

exceeding languages produced by these two nets is quite similar, while the distances are not; the cause is that the distance, differently from the exceeding language, is a global measure which depends from the initial marking and also all the markings reachable from it. As a consequence, deviating behaviours that the exceeding language can show only for great values of the length  $n$  are immediately represented for smaller distance lengths. For example,  $\langle N'', \bar{m}''_0 \rangle$  reaches the illegal observable marking  $\bar{m} = [1100]^T$  by firing the sequence of transitions  $\sigma = t_1 t_2 t_4 t_5$ ; this is represented by  $d^1(N'', \bar{m}''_0) > 0$ , but requires a length  $n \geq 5$  in order to be shown up in  $L_{Exc}^n(N'', \bar{m}''_0)$ .

To conclude, the resulting model obtained with the pro-

posed approach is a general net, while previous solutions only return 1-bounded nets. However, the proposed approach works on logical models, while the explicit consideration of time is becoming crucial for the specification and verification of systems such as transportation systems [4], real-time systems, as well as the study of problems such as state estimation, and fault diagnosis [1]. First results to the identification of timed net models have been proposed in [3], [2] to identify a Time Petri Net from sequences of timed transition firings; future research will be devoted also on the identification of timed IPN models.

## REFERENCES

- [1] F. Basile, M. P. Cabasino, and C. Seatzu. Diagnosability analysis of labeled time Petri net systems. *IEEE Trans. Autom. Control*, 62(3):1384–1396, March 2017.
- [2] F. Basile, P. Chiacchio, and J. Coppola. A Novel Model Repair Approach of Timed Discrete-Event Systems With Anomalies. *IEEE Transactions on Automation Science and Engineering*, 13(4):1541–1556, October 2016.
- [3] F. Basile, P. Chiacchio, and J. Coppola. Identification of time petri net models. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(9):2586–2600, Sep. 2017.
- [4] F. Basile, P. Chiacchio, and D. Teta. A hybrid model for real time simulation of urban traffic. *Control Engineering Practice*, 20(2):123 – 137, 2012.
- [5] M.P. Cabasino, P. Darondeau, M.P. Fanti, and C. Seatzu. Model identification and synthesis of discrete-event systems. In Mengchu Zhou, Han-Xiong Li, and Margot Weijnen, editors, *Contemporary Issues in Systems Science and Engineering*, IEEE/Wiley Press Book Series, pages 343–366. John Wiley & Sons, Inc., 2015.
- [6] M.P. Cabasino, A. Giua, and C. Seatzu. Identification of Petri nets from knowledge of their language. *Discrete Event Dynamic Systems*, 17:447–474, December 2007.
- [7] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Deriving Petri nets from finite transition systems. *IEEE Trans. on Computers*, 47(8):859–852, August 1998.
- [8] P. Darondeau. Region based synthesis of p/t-nets and its potential applications. In *21st Int. Conf. on App. and Theory of Petri Nets 2000 (ICATPN 2000) Aarhus, Denmark*, volume 1825 of *Lecture Notes in Computer Science*, pages 16–23. Springer, June 2000.
- [9] M. Dotoli, M.P. Fanti, and A.M. Mangini. Real time identification of discrete event systems using Petri nets. *Automatica*, 44(5):1209 – 1219, 2008.
- [10] A. P. Estrada-Vargas, J. Lesage, and E. Lopez-Mellado. Identification of industrial automation systems: Building compact and expressive petri net models from observable behavior. In *2012 American Control Conference (ACC)*, pages 6095–6101, June 2012.
- [11] A. P. Estrada-Vargas, E. Lopez-Mellado, and J. Lesage. Identification of partially observable discrete event manufacturing systems. In *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–7, Sept 2013.
- [12] A. P. Estrada-Vargas, E. Lopez-Mellado, and J. Lesage. A black-box identification method for automated discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 14(3):1321–1336, July 2017.
- [13] A. Ghaffari, N. Rezg, and Xiaolan Xie. Design of a live and maximally permissive petri net controller using the theory of regions. *IEEE Transactions on Robotics and Automation*, 19(1):137–141, Feb 2003.
- [14] T. Murata. Petri nets: Properties, analysis and applications. *77(4):541–580*, April 1989.
- [15] J. Saives, G. Faraut, and J. Lesage. Identification of discrete event systems unobservable behaviour by petri nets using language projections. In *2015 European Control Conference*, pages 464–471, 2015.
- [16] Ana Paula Estrada Vargas. *Black-box Identification of Automated Discrete Event Systems*. PhD. Centro de Investigacin y de Estudios Avanzados del I.P.N., 2013.
- [17] G. Zhu, Z. Li, and N. Wu. Model-based fault identification of discrete event systems using partially observed petri nets. *Automatica*, 96:201 – 212, 2018.