Identification of Time Petri Net models

Francesco Basile, Pasquale Chiacchio, Jolanda Coppola

***Abstract*— This paper deals with the identification of Time Petri net systems. An identification algorithm for timed net systems must take into account that the firing of a transition requires not only that the enabling condition is met, as in untimed net systems, but it is also required that the firing interval of a transition is congruent with the observed firing instant times. The key idea behind the approach is to express these conditions by a set of logical propositions that can be directly transformed into linear mixed-integer inequalities. The identification algorithm consists of building the logical propositions from the observed behavior and solving a mixed-integer linear programming problem.**

I. INTRODUCTION

A. Position of the paper

System identification of discrete event processes/systems from external observation of their behavior is a challenging problem that received a lot of attention in the last decade.

The interest for the identification of Discrete Event Systems (DESs) usually comes from reverse engineering for (partially) unknown systems, fault diagnosis, or system verification . Input and/or output sequences are observed during the operation of the system within its environment. 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 behavior from sequences observed during the system operation [1], [2]. When the resulting model is a PN, the net structure (places, transitions and arcs) and its initial marking must be identified.

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 [3], [4], [5], [6]. If this is the case, the tackled problem is more a *net synthesis* problem, rather than a *net identification* one. When dealing with net synthesis, the net system is typically built *offline* starting from the available data.

When a set of observed strings, i.e., a subset of the system language, and/or a set of observed net markings are available, the related problem is a proper *net identification* problem [7], [8], [9]. In such a framework, the periodical execution of an identification algorithm that provides a model able to generate the observed strings is the main goal.

Francesco Basile, Pasquale Chiacchio and Jolanda Coppola are with Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata, Università di Salerno, Via Giovanni Paolo II, 132 - 84084 Fisciano (SA) - Italy. Corresponding author email: fbasile@unisa.it

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Postprint - Work published on IEEE Transactions on System, Man and Cybernetics (http://dx.doi.org/10.1109/TSMC.2016.2523929)

Both net identification and net synthesis are very complex problems. Restricting net subclass, efficient results can be obtained as in [10] where safe Interpreted PNs are identified in polynomial time.

Process mining [11], [12] aims to discover, monitor and improve real processes by extracting knowledge from a collection of sequential events and information about the system. This problem is strongly related to the identification problem treated by Discrete Event System community. In [13] an algorithm to mine (identify) a workflow is presented. Such an algorithm allows to identify a particular class of PN, called workflow net, by analyzing the direct casuality between firings of transitions. A workflow net is an acyclic net with one source place, called an input place, and one sink place, called an output place, strongly connected and sound (i.e., it is live and safe).

The explicit consideration of time is crucial for the specification and the verification of some DESs such as communication protocols, circuits, real-time systems, automated warehouse systems [14], urban traffic systems [15], automated manufacturing systems [16]. Two main techniques were developed from PNs to associate a timing structure to transitions: Timed PNs [17] and Time PNs (TPNs) [18]. In the first one, a fixed firing duration is associated with each transition while, in the second one, the firing duration of a transition t can assume any value of a given interval $I(t)$. Two other techniques are available to deal with time intervals: P-TPN [19] and Arc-TPNs [20]. In P-TPNs the timed behavior is implemented by means of the tokens in the net (each token has its own age) and the time intervals associated to the places, which restrict the age of tokens that can be used in order to fire the output transition from the places. In Arc-TPNs the timed behavior is implemented by means of the tokens in the net (each token has its own age) and the time intervals associated to arcs from places to transitions, which restrict the age of tokens that can be used in order to fire the respective transition. Both P-TPN and Arc-TPNs model the death of tokens, i.e., a token is too old to enable a transition, and the dependency of transition firings on local clocks (the age of tokens that enable it), so they are suitable to application in the field of batch process control. In [21] a nice comparison of the expressiveness power of these techniques is presented.

In [9] the identification of the net structure of a deterministic timed PN system is treated. The firing duration of transitions is assumed to be known and the proposed algorithm identifies the structure and the initial marking of the timed PN. Timing information is used to accelerate the net identification with respect to the classical untimed approaches.

In [22] the identification of the net structure and the initial marking is carried out at a first stage and then the timing structure is inferred from additional observations, using the net structure identified at the first stage. Stochastic and deterministic stochastic PNs with deterministic and exponentially distributed transition firing durations are considered. In [23] the identification is considered for timed interpreted PN systems, a net subclass where the timing structure represents the time interval during which a token remains available in a place. The net marking is assumed to be observable in addition to the event occurrences. The timing structure is inferred from the observation of the net marking. Then, even if both the timing structure and the net structure are identified, the identification can be seen as a two stage identification process.

B. Contribution of the paper

This paper focuses on the identification of TPN, which can model manufacturing systems where the age of tokens is not relevant, as for example assembly processes and material handling systems, from a set of observed timed sequences. No specific assumptions are made about the topology of the identified net system, its net structure is not required to belong to a specific subclass and may be cyclic.

The goal of this paper is not only the identification of the net structure and the initial marking but the identification of the firing duration of transitions (timing structure) as well.

To the best of the authors' knowledge, only few works have been published on this topic.

Given a set of observed timed sequences (i.e., sequences of transition firings with their firing time instants), a number of places m and a set of transitions T, the considered identification problem consists in determining the structure of a net N, i.e., the pre and post incidence matrices $\mathbf{Pre}$ and $\mathbf{Post}$ and its initial marking $\boldsymbol{m}_0$, the timing structure $I(t)$, i.e., the firing interval of each transition $t \in T$, such that the set of timed sequences generated by this net system contains all the observed ones. As usual in DES identification, the identified system can also produce timed sequences that do not belong to the unknown system behavior, as well as, it can reproduce sequences of the original system that have not been observed.

The presence of a timing structure implies that when a transition t fires at a time τ, i) it has been enabled at a time $\tau' \leq \tau$ ($=$ stays for immediate firings) and ii) its firing duration $\delta = \tau - \tau'$ is congruent with the observation, i.e., $\delta \in I(t)$. Enabling of transitions and their firing time instants are characterized in terms of mixed-integer linear constraints as in [3].

To accelerate the identification, an estimation of the timing structure is assumed to be known. More precisely, the mimimal lower and the maximal upper bound of transition firing intervals are assumed to be known.

The use of the timing information allows to accelerate the identification process with respect to the untimed approach. Indeed, the set of markings that can enable an observed firing of a transitions is reduced, since markings reached at time smaller than the lower bound or greater than the upper bound must not be considered.

The identification process reduces to a single stage, using a unique set of observations, proving to be more convenient than a two stage approach that works on a complete (or partial) knowledge of the system language to identify the net structure and afterwards to infer the time duration of the transitions from the timed sequences.

Moreover, the proposed approach works on effective observations produced by these systems considering that two events can occur at the same time. In untimed/logical PN models it is assumed the occurrence of two events cannot happen simultaneously [24], [25] even if they model concurrent activities with no causal relationship.

This paper focuses on the context of automated manufacturing systems, where a controlling agent interacts with a plant. In this context, some events are managed by the control architecture, enabling or forcing their occurrence by sending proper signals to the actuators in a plant. These events, called *controllable* events, are usually known, being the outputs of the controlling agent that is usually accessible, and have null firing duration, i.e., they are *immediate* events.

For selecting among different solutions, a performance index is used. As for example, in this paper a very general index involving the arc weights, the number of tokens in the initial marking and the firing interval of transitions, is minimized. However, different choices can be made for the cost function depending on the context.

A preliminary version of this approach has been presented in [26] where the authors chose a different treatment of timed and immediate transitions and assumed to know the whole set of immediate transitions and their firing order. Here, timed and immediate transitions are treated in the same way, but a distinction between immediate and timed firings of a transition is made. In particular, immediate firings include the firings of immediate transitions and the firings of timed transitions with null firing duration. Finally, the classification of transitions in controllable and uncontrollable ones is introduced. The first ones, as stated above, are assumed to be only immediate and their set is assumed to be known; the second ones can be timed as well as immediate - no preliminary information about them is given.

The motivation for this research comes also from two interesting application domain, fault detection and model repair, that are briefly recalled in the following.

Fault detection of discrete event systems is usually carried out using models that included the faulty behavior [27]. From a practical point of view, a fault-free model would have much more sense, but in this case fault isolation and identification may not be possible because the model does not include the faulty behaviour and, therefore, the diagnosability of a given fault is not guaranteed. However, one may remove the requirement that the nature (or behavior) of the faults is known, and this is realistic, but at the same time an algorithm could be used to identify the faulty behavior, so obtaining a faulty model from real data. PNs and their extension are largely used in this context [27], [28]. A fault is detected if an observed behavior of the system cannot be reproduced by its model. The approach proposed in this paper may be used to identify the timed faulty model (a model that includes the effects of timed faults) adding to the fault-free model a new subsystem that is able to include the faulty observed behavior, not included in the system nominal language. A similar approach has been used in [29] for untimed systems, and in [30] to obtain a fault detection and recovery strategy in robot task executions, using an untimed system too. The availability of a timed faulty model of a system can accelerate significantly the fault diagnosis with respect to the untimed case.

The approach presented in this paper may be adapted also to identify system anomalies in order to obtain a "repaired model" where also discrepancies about activities duration are considered. Model repair is a relevant problem that has received a lot of attention in the field of Workflow Management Systems community, where it has been treated as a part of Process mining problem. It consists in modifying the nominal model of a system as consequence of the occurrence of discrepancies (named anomalies) between the system nominal behavior and the system observed behavior [**?**]. As stated in [31] detection and automated modeling of anomalies can simplify the task of finding potential errors in business processes.

II. PN BACKGROUND

In this section, a brief recall of the PNs and Time PNs theory is done. For a complete review on PNs the reader can refer to [32].

A *Place/Transition* net (*P/T* net) is a 4-tuple $N = (P, T, \mathbf{Pre}, \mathbf{Post})$, where P is a set of m places (represented by circles), T is a set of n transitions (represented by boxes), $\mathbf{Pre} : P \times T \to \mathbb{N}$ ($\mathbf{Post} : P \times T \to \mathbb{N}$) is the *pre* (*post*) *incidence* matrix. $\mathbf{Pre}(p,t) = w$ ($\mathbf{Post}(p,t) = w$) means that there is an arc with weight w from p to t (from t to p); $\boldsymbol{C} = \mathbf{Post} - \mathbf{Pre}$ is the incidence matrix. The symbols ${}^\bullet p$ (${}^\bullet t$) and $p^\bullet$ ($t^\bullet$) are used for the *pre-set* and *post-set* of a place $p \in P$ (transition $t \in T$), respectively, e.g. ${}^\bullet t = \{p \in P \mid \mathbf{Pre}(p,t) \neq 0\}$.

A *marking* is a function $\boldsymbol{m} : P \to \mathbb{N}$ that assigns to each place of a net a nonnegative integer number of tokens, drawn as black dots. It is useful to represent the marking of a net with a vector $\boldsymbol{m} \in \mathbb{N}^m$. A *net system* $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$ is a net N with an initial marking $\boldsymbol{m}_0$. A transition t is enabled at $\boldsymbol{m}$ iff $\boldsymbol{m} \geq \mathbf{Pre}(\cdot, t)$ and this is denoted as $\boldsymbol{m}[t\rangle$. An enabled transition t may fire yielding the marking $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C}(\cdot, t)$ and this is denoted as $\boldsymbol{m}[t\rangle \boldsymbol{m}'$.

A *firing sequence* from $\boldsymbol{m}$ is a sequence of transitions $\sigma = t_1 \ldots t_k$ such that $\boldsymbol{m}[t_1\rangle \boldsymbol{m}_1 [t_2\rangle \boldsymbol{m}_2 \ \ldots [t_k\rangle \boldsymbol{m}_k$, and this is denoted as $\boldsymbol{m}[\sigma\rangle \boldsymbol{m}_k$. An enabled sequence σ is denoted as $\boldsymbol{m}[\sigma\rangle$, while $t_i \in \sigma$ denotes that transition t_i belongs to the sequence σ. A marking $\boldsymbol{m}'$ is said to be *reachable* from $\boldsymbol{m}_0$ iff there exists a sequence σ such that $\boldsymbol{m}_0[\sigma\rangle \boldsymbol{m}'$. $R(N, \boldsymbol{m}_0)$ denotes the set of reachable markings of the net system $\langle N, \boldsymbol{m}_0 \rangle$.

Given a sequence σ it is denoted with $|\sigma|$ its length.

The function $\boldsymbol{\sigma} : T \to \mathbb{N}$, where $\boldsymbol{\sigma}(t)$ represents the number of occurrences of t in σ, is called *firing count vector* of the firing sequence σ. As it has been done for the marking of a net, the firing count vector is often denoted as a vector $\boldsymbol{\sigma} \in \mathbb{N}^n$. Note that, if a sequence is made up of a single transition, i.e., $\sigma = t_i$, then the corresponding firing count vector is the i-th canonical basis vector denoted as $\boldsymbol{e}_i$.

If $\boldsymbol{m}_0[\sigma\rangle \boldsymbol{m}$, then it is possible to write in vector form

$$\boldsymbol{m} = \boldsymbol{m}_0 + \big(\mathbf{Post} - \mathbf{Pre}\big) \cdot \boldsymbol{\sigma} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}\,, \qquad (1)$$

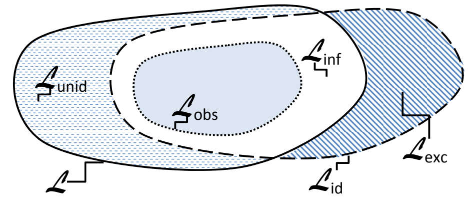

Fig. 1. Graphical representation of the different considered languages.

which is called the *state equation* of the net system.

Definition 1 (Time Petri net system, [33]): Let $\mathcal{I}$ be the set of closed intervals with a lower bound in $\mathcal{Q}$ and an upper bound in $\mathcal{Q} \bigcup \infty$. A *Time* Petri net (TPN) system is the 3-ple $\mathcal{T} = \langle N, \boldsymbol{m}_0, I \rangle$, where N is a standard P/T net, $\boldsymbol{m}_0$ is the initial marking, and $I : T \to \mathcal{I}$ is the *statical firing time interval function* which assigns a firing interval $[l_j, u_j]$ to each transition $t_j \in T$.

It is assumed that there is a start-up transition that fires only once at time zero producing tokens considered by the initial marking. A transition t_j can be fired at time τ if the time elapsed from the enabling belongs to the interval $I(t_j)$; moreover, when enabled, t_j must fire if the upper bound of $I(t_j)$ is reached, thus enforcing urgency. If $l_j = u_j = 0$ (and consequently $I(t_j) = [0,0]$) transition t_j is said *immediate* otherwise it is said *timed*. Timed and immediate transitions will be represented by empty and filled boxes, respectively. $\Diamond$

Given a set S, $|S|$ denotes the cardinality of S. In the following T^t is denoted as the set of timed transitions, with cardinality $n_t = |T^t|$ and T^{im} is denoted as the set of immediate transitions, with cardinality $n_{im} = |T^{im}|$.

Definition 2 (Timed firing sequence): A sequence

$$\sigma_T = (T_1, \tau_1) \ldots (T_q, \tau_q) \ldots (T_L, \tau_L)\,,$$

where T_q is the set of transitions fired at time τ_q, $\tau_1 < \tau_2 \cdots < \tau_L$ denote firing time instants, is called *timed firing sequence*. The position q the couple (T_q, τ_q) occupies in the sequence is called *time step*, so (T_1, τ_1) is associated with step 1, (T_2, τ_2) is associated with step 2 and so on; the number of couples (T_q, τ_q) in σ_T is called length $L = |\sigma_T|$ of the timed firing sequence.

The notation $\boldsymbol{m}[(T_q, \tau_q)\rangle \boldsymbol{m}'$ denotes that $\boldsymbol{m}'$ is reached from $\boldsymbol{m}$ by firing the transitions in the set T_q at time τ_q. The notation $\boldsymbol{m}[\sigma_T\rangle \boldsymbol{m}'$ denotes that $\boldsymbol{m}'$ is reached from $\boldsymbol{m}$ by firing σ_T. Moreover, the notation $\boldsymbol{m}[T_q\rangle \boldsymbol{m}'$ denotes that $\boldsymbol{m}'$ is reached from $\boldsymbol{m}$ by firing the transitions in the set T_q at the same time but without referring to any specific time. $\Diamond$

Definition 3 (Timed Language): Given a TPN system $\mathcal{T} = \langle N, \boldsymbol{m}_0, I \rangle$, its timed language, named $L(\mathcal{T})$, is defined as the set of timed firing sequences generated by $\mathcal{T}$ from the initial marking $\boldsymbol{m}_0$. $\Diamond$

III. IDENTIFICATION OF TIMED DESS

The goal of this paper is the *identification* of a TPN model that is able to generate the observed language of an unknown timed system, starting from a subset of the language the system can generate. Inspired by [34], the relation between

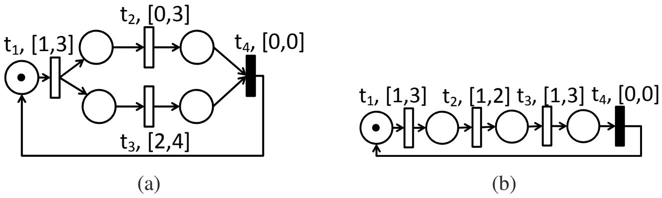

Fig. 2. Unknown (a) and identified system (b) on the basis of the observed language $\mathcal{L}_{obs} = \{\sigma_T = (\{t_1\},1)(\{t_2\},2)(\{t_3,t_4\},3)(\{t_1\},6)(\{t_2\},8)(\{t_3,t_4\},10)\}$.

the unknown system language, the observed language and the identified language is graphically represented in Fig. 1.

The complete language of the unknown system, namely $\mathcal{L}$ in Fig. 1, can be divided in two subsets of sequences: the subset of the observed timed sequences, which represents the *observed language* of the system, namely $\mathcal{L}_{obs}$, and the subset of the unobserved ones, which represents the *unobserved language* of the system, namely $\mathcal{L}_{unobs} = \mathcal{L} \setminus \mathcal{L}_{obs}$.

The considered identification problem consists in determining the structure of a time PN, i.e., the matrices $\mathbf{Pre}$, $\mathbf{Post}$ and its initial marking $\boldsymbol{m}_0$, the timing structure $I(t),\ \forall t \in T$ such that the set of timed sequences generated by this net, called $\mathcal{L}_{id}$ in Fig. 1, is such that $\mathcal{L}_{id} \supseteq \mathcal{L}_{obs}$.

The language of the identified system, in general, contains a subset of timed sequences that belong to $\mathcal{L}_{id}$ but do not belong to $\mathcal{L}$. Such a subset represents the *exceeding language* of the identified system and it is indicated as $\mathcal{L}_{exc} = \mathcal{L}_{id} \setminus \mathcal{L}$ in Fig. 1.

The language of the unknown system, in general, contains a subset of timed sequences that belong to the unknown system but not to the identified one. Such a subset represents the *unidentified language* of the unknown system and it is indicated as $\mathcal{L}_{unid} = \mathcal{L} \setminus \mathcal{L}_{id}$ in Fig. 1.

As example, consider the unknown system shown in Fig. 2(a) and assume that $\mathcal{L}_{obs} = \{\sigma_T = (\{t_1\},1)(\{t_2\},2)(\{t_3,t_4\},3)(\{t_1\},6)(\{t_2\},8)(\{t_3,t_4\},10)\}$. The identified net is the one shown in Fig. 2(b).

For these nets it is possible to find at least one timed sequence that belongs to $\mathcal{L}_{unid}$ as well as one timed sequence that belongs to $\mathcal{L}_{exc}$. As for example, the sequence $\sigma'_T = (\{t_1\},1)\ (\{t_3\},3)\ (\{\{t_2,t_4\},4) \in \mathcal{L}$ does not belong to $\mathcal{L}_{id}$, i.e., $\sigma'_T \in \mathcal{L}_{unid}$, while the sequence $\sigma''_T = (\{t_1\},3)\ (\{t_2\},5)\ (\{t_3,t_4\},8) \in \mathcal{L}_{id}$ does not belong to $\mathcal{L}$, i.e., $\sigma''_T \in \mathcal{L}_{exc}$.

Moreover, the language $\mathcal{L}_{id}$ also contains a set of sequences that belong to $\mathcal{L} \bigcap \mathcal{L}_{id}$ but do not belong to $\mathcal{L}_{obs}$. Such a set represents the *inferred language* of the unknown system and it is indicated as $\mathcal{L}_{inf} = \mathcal{L} \bigcap \mathcal{L}_{id} \setminus \mathcal{L}_{obs}$ in Fig. 1. This means that the identified system is able to reproduce also sequences of the unknown system language without directly observing them. With reference to the previous example, $\sigma'''_T = (\{t_1\},2)(\{t_2\},3)(\{t_3,t_4\},5) \in \mathcal{L}_{inf}$.

A desirable characteristic of an identification algorithm is to obtain a language $\mathcal{L}_{inf}$ as larger as possible with a language $\mathcal{L}_{obs}$ as smaller as possible.

An identification algorithm returns a model of the system every time a new observation is available, in contrast with approaches that can work only when all the observations the system can generate are available.

Each time a new timed sequence σ_T is observed, it is added to the observed language and it is tested via simulation if such a sequence belongs to the language of the system that is currently identified: if so, the algorithm waits for a new observation, otherwise a new identification problem, which takes into account the updated observed language, is solved. The solution of the problem is assumed as the current identified model of the system. Then a new observation can be considered.

However, if the system dynamic is too fast, i.e., the time required to process the event observed at a given time instant is great with respect to the event interarrival time, observation and identification problem resolution can be also executed separately, processing the observation in batches : it is possible to collect a certain number of timed sequences observed in real time from the unknown system behavior and then to solve the corresponding identification problem to obtain the new model.

IV. Assumptions

In this section, assumptions and definitions, needed for developing the proposed net identification approach, are presented.

Assumption 1 (Properties of the unknown system): The observed system can be modeled by a TPN system with the following assumptions

1) *Free labeled nets*, i.e., there is an isomorphism between the label set E and the transition set T. Observing the evolution of a net, it is common to assume that a label is assigned to each transition t, and the firing of t is an event that generates the label as observable output. This assumption restricts the net subclass that can be identified by the proposed approach to free labeled nets, but it allows to speak of event observation as well as of firing of transitions without any difference. Moreover, it implies that the firing of each transition can be directly observed.
2) *k-bounded nets*, i.e., the number of tokens in each place of the net is never greater than k.
3) *Single-server* firing semantic (more details in [33]), i.e., no concurrent firings of the same transition are possible.
4) *Enabling memory* policy of timed transitions, i.e., when a new marking is reached and a timed transition is not enabled, the elapsed time is reset. $\Diamond$

The transition set T is partitioned into the set T^c of controllable transitions, with cardinality n_c, and the set T^{uc} of uncontrollable transitions, with cardinality n_{uc}.

Assumption 2 (Controllable transitions): It is assumed that:

1) All controllable transitions are known and immediate since they are managed by the controller.
2) All transitions that make up a choice – i.e., all transitions $t \in p^\bullet$ with $|p^\bullet| > 1$ – must be controllable. Hence if $|p^\bullet| > 1 \quad \Rightarrow \quad I(t) = [0,0]\ \forall\ t \in p^\bullet$, for all the places in P. This assumption is motivated by the consideration that, when a timed activity is associated

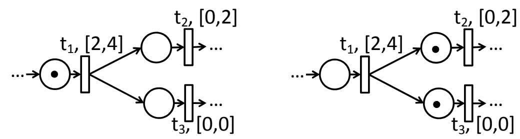

Fig. 3. Evolution of the net system of Example 1.

with conflicting transitions, a conflict resolution policy may be a race between conflicting transitions, which is pointless in the context of manufacturing systems. Then, conflicts only include controllable transitions which are immediate and known, so a set of constraints, which must be fulfilled in order to guarantee this assumption holds, can be devised. It is presented in Section V-B. ◊

The approach presented in this paper does not perform any controlling action on the system by means of controllable events, but only assumes that the set of controllable events is known. This is realistic in the context of manufacturing systems, where controllable events are the outputs of the controlling agent, which is usually accessible.

Assumption 3: A transition can fire only once at the same time instant. ◊

This assumption is motivated by the consideration that the multiple firings of a transition at the same time are pointless in the context of manufacturing systems where, in practice, the interaction between the plant and the controller occurs according to a scan time faster than the evolution time of the system. However, the results presented in this paper are still valid removing this assumption, introducing some technicalities.

For a better presentation of the approach proposed in this paper it is useful to collect all transitions that fire at the same time τ in the same set.

The set T_q is made up of $n_q = |T_q|$ transitions whose firing is observed at the same instant τ_q. The marking the system reaches after the firing of all the transitions in T_q is called $\boldsymbol{m}_q$. The firings of these transitions are enabled either by a marking $\boldsymbol{m}_k$, reached at a time $\tau_k < \tau_q$, or by the firing of another transition fired at τ_q with null firing duration.

Definition 4 (Firing Interval): Given a timed transition t_j fired at q-th step, enabled at k-th step, so that $\boldsymbol{m}_k[t_j\rangle$, let $\boldsymbol{m}_k$ be the first marking that enables t_j since its previous firing, the function $\delta(t_j, k, q) : T \times \mathbb{N} \times \mathbb{N} \to \mathcal{Q}$ returns the time elapsed from the enabling of t_j at τ_k until its firing at τ_q, i.e., $\delta(t_j, k, q) = \tau_q - \tau_k$. ◊

From now on, $\delta(t_j, k, q)$ is referred as the firing duration of $t_j \in T^t$ from the marking $\boldsymbol{m}_k$. When $\delta(t_j, k, q) = 0$ the firing of t_j at τ_q is called *immediate*, otherwise, when $\delta(t_j, k, q) > 0$ it is called *timed*.

In the following, two simple examples are discussed to motivate the partition of T_q in two disjoint sets, the set T_q^t of transitions with timed firings and the set T_q^{im} of transitions with immediate firings at time τ_q, as well as a further partition of the last one according to a firing order. Obviously, since the firing of a controllable transition is always immediate, each $t_j \in T^c \cap T_q$ will belong to T_q^{im}. For the sake of simplicity, in the next two examples controllable transitions are not considered, i.e., all transitions are assumed uncontrollable, since the scope of these examples is to discuss some differences between timed and immediate firings.

Example 1: In this example it is discussed that immediate firings of transitions in the set T_q always come after the timed ones, even if they are observed at the same instant.

Consider the system shown in Fig. 3(a) at time τ_0: under the current net marking, named $\boldsymbol{m}_0$, t_1 fires at a time $\tau_1 \in [\tau_0 + 2, \tau_0 + 4]$ and the marking $\boldsymbol{m}_1$, shown in Fig. 3(b), is reached, where both t_2 and t_3 are enabled. Obviously t_3 fires at τ_1, since it is immediate. If t_2 fires at τ_1 too, the firing duration of t_2 from $\boldsymbol{m}_1$ is $\delta(t_2, 1, 1) = 0$, and the marking $\boldsymbol{m}_2$ is reached. Marking $\boldsymbol{m}_1$ results to be a *vanishing marking*[1], since it enables an immediate firing. The marking $\boldsymbol{m}_0$ is said *tangible marking* because only timed firings are enabled.

The timed firing of t_1 precedes the immediate firings of t_2 and t_3, but the firings of t_1, t_2 and t_3 are observed at the same time and so they belong to the same set T_1. Moreover, it is not possible to know *a priori* the firing order of the transitions from the observation. ◊

Example 2: In this example it is discussed that transitions firing at the same instant, with timed firings, can have been enabled by different tangible markings and that the immediate firings of transitions can occur sequentially even if they have been observed at the same time.

Consider the net system in Fig. 4(a) at time τ_0. Under the initial net marking, named $\boldsymbol{m}_0$, both t_1 and t_4 are enabled. Assume transition t_4 fires at $\tau_1 = \tau_0 + 1$, then $\boldsymbol{m}_1$ is reached (see Fig. 4(b)) where t_5 is enabled. If the firing duration of t_1 from $\boldsymbol{m}_0$ is equal to 2, at time $\tau_2 = \tau_1 + 1 = \tau_0 + 2$ both t_1 and t_5 fires, leading to the marking $\boldsymbol{m}_2$ (Fig. 4(c)). The marking $\boldsymbol{m}_2$ results to be vanishing since t_2 is immediate and fires immediately after t_1 at time $\tau_3 = \tau_2$, leading the system in the marking $\boldsymbol{m}_3$ where t_3 is enabled. Finally, suppose that t_3 fires at $\tau_4 = \tau_3 = \tau_2$, i.e., $\delta(t_3, 3, 4) = 0$. The timed sequence $(\{t_4\}, \tau_1)(\{t_1, t_2, t_3, t_5\}, \tau_2)$ is observed. As it is shown in Fig. 4(d), there are two different tangible markings, $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$, that enable two timed firings at τ_2, so each timed firing in T_q can be enabled at a different tangible marking. Moreover, immediate firings observed at τ_2 are enabled at the two vanishing markings $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$, resulting to be sequential even if they are observed simultaneously. ◊

Let $\boldsymbol{m}_0$ be the initial marking of the system, the set of candidate markings for the enabling of a transition $t_j \in T_q$ can be formally defined as $\mathcal{M}(t_j, q) = \{\boldsymbol{m}_k \mid \exists \sigma_T^{'}, \sigma_T^{''}, \sigma_T = \sigma_T^{'}\sigma_T^{''}, \ \boldsymbol{m}_0[\sigma_T^{'}\rangle \boldsymbol{m}_k[\sigma_T^{''}\rangle \boldsymbol{m}_q, \text{ with } t_j \in \sigma_T^{''}, \ k < q \ : \ \tau_k + l_j \leq \tau_q \leq \tau_k + u_j\}$, having cardinality $|\mathcal{M}(t_j, q)|$.

The set T_q can be partitioned into the couple (T_q^t, T_q^{im}): $T_q^t = \{t_j \in T_q | \ \exists k, \boldsymbol{m}_k \in \mathcal{M}(t_j, q)\}$ is the set of transitions fired at τ_q with timed firing, with cardinality $n_q^t = |T_q^t|$, $T_q^{im} = T_q \setminus T_q^t$, with cardinality n_q^{im}, is the set of transition fired at τ_q with immediate firing.

With reference to the example of Fig. 4, $T_2^t = \{t_1, t_5\}$, and $T_2^{im} = \{t_2, t_3\}$.

[1] A *vanishing marking* is a marking in which at least one immediate transition is enabled, otherwise the marking is called *tangible marking* [35].

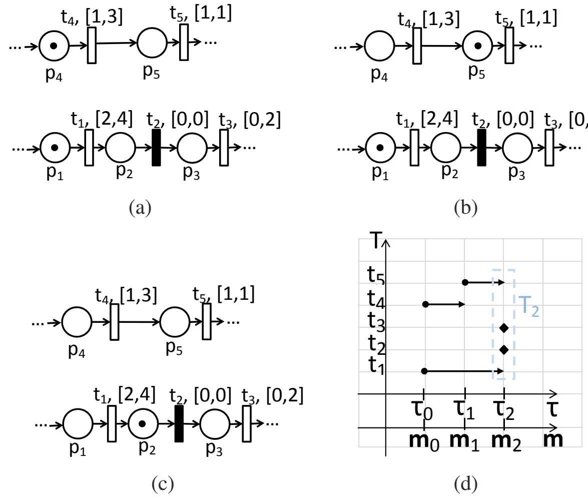

Fig. 4. (a)-(c) Evolution of the net system of Example 2; (d) enabling and firing time of each transition of the net: for timed firings of transitions, dots represent the enabling instant, the arrow points individuate the firing instant and the length of the arrows coincides with the firing duration value; diamonds individuate enabling and firing instant for immediate firings of the transitions.

However, on the basis of the observed firing sequence, the composition of T_q^t and T_q^{im} is unknown and all the possible combinations must be considered. Denote $(T_{q,h}^t, T_{q,h}^{im})$ the h-th possible couple (T_q^t, T_q^{im}). With reference to the example of Fig. 4, the following cases are allowed:

- $T_{2,1}^t = \{t_1\}$, $T_{2,1}^{im} = \{t_2, t_3, t_5\}$;
- $T_{2,2}^t = \{t_2\}$, $T_{2,2}^{im} = \{t_1, t_3, t_5\}$;
- $T_{2,3}^t = \{t_3\}$, $T_{2,3}^{im} = \{t_2, t_1, t_5\}$;
- $T_{2,4}^t = \{t_5\}$, $T_{2,4}^{im} = \{t_2, t_3, t_1\}$;
- $T_{2,5}^t = \{t_1, t_2\}$, $T_{2,5}^{im} = \{t_3, t_5\}$;
- $T_{2,6}^t = \{t_1, t_3\}$, $T_{2,6}^{im} = \{t_2, t_5\}$;
- $T_{2,7}^t = \{t_1, t_5\}$, $T_{2,7}^{im} = \{t_2, t_3\}$;
- $T_{2,8}^t = \{t_2, t_3\}$, $T_{2,8}^{im} = \{t_1, t_5\}$;
- $T_{2,9}^t = \{t_2, t_5\}$, $T_{2,9}^{im} = \{t_1, t_3\}$;
- $T_{2,10}^t = \{t_3, t_5\}$, $T_{2,10}^{im} = \{t_1, t_2\}$;
- $T_{2,11}^t = \{t_1, t_2, t_3\}$, $T_{2,11}^{im} = \{t_5\}$;
- $T_{2,12}^t = \{t_1, t_2, t_5\}$, $T_{2,12}^{im} = \{t_3\}$;
- $T_{2,13}^t = \{t_1, t_3, t_5\}$, $T_{2,13}^{im} = \{t_2\}$;
- $T_{2,14}^t = \{t_2, t_3, t_5\}$, $T_{2,14}^{im} = \{t_1\}$;
- $T_{2,15}^t = \{t_1, t_2, t_3, t_5\}$, $T_{2,15}^{im} = \emptyset$.

In general, the number of all possible couples $(T_{q,h}^t, T_{q,h}^{im})$ is $c_q = \sum_{i=1}^{n_q} \binom{n_q}{i} = \sum_{i=1}^{n_q} \frac{n_q!}{i!(n_q-i)!}$.

All the transitions of $T_{q,h}^t$ fire concurrently. Denote as $\boldsymbol{m}_{q_1}$ the marking reached by firing the transitions belonging to $T_{q,h}^t$, i.e., with reference to the example of Fig. 4, after the firing of transitions $\{t_1, t_5\} \in T_2^t$ marking $\boldsymbol{m}_{2_1}$ is reached. Notice that the marking $\boldsymbol{m}_{q_1}$ is vanishing if $T_{q,h}^{im}$ is not empty.

On the other hand, transitions in $T_{q,h}^{im}$ may fire sequentially. This means that given the set of transitions $T_{q,h}^{im}$, these transitions can fire in any order, included concurrently. Denote as $\boldsymbol{m}_{q_s}$, with $s \geq 2$ the vanishing marking reached after the immediate firings of transitions in $T_{q,h}^{im}$, then, with reference to the example of Fig. 4, considering $T_{2,7}^{im} = \{t_2, t_3\}$, the following firing sequences may be allowed:

$\boldsymbol{m}_{2_1}[t_2\rangle \boldsymbol{m}_{2_2}[t_3\rangle \boldsymbol{m}_{2_3}$;
$\boldsymbol{m}_{2_1}[t_3\rangle \boldsymbol{m}_{2_2}[t_2\rangle \boldsymbol{m}_{2_3}$;
$\boldsymbol{m}_{2_1}[\{t_2, t_3\}\rangle \boldsymbol{m}_{2_2}$.

The number of all possible firing sequences is $c_{q,h} = (n_{q,h}^{im})! + \sum_{j=2}^{n_{q,h}^{im}} \binom{n_{q,h}^{im}}{j} \cdot (n_{q,h}^{im} - j + 1)!$, where $n_{q,h}^{im}$ is the cardinality of $T_{q,h}^{im}$. The length of such a sequences is 1 when all transitions in $T_{q,h}^{im}$ fire concurrently, while it is $n_{q,h}^{im}$ when all the firings are sequential. Then, the firing of the transitions in $T_{q,h}^{im}$ can be considered as a sequence of concurrent transition firings that define subsets of $T_{q,h}^{im}$.

Given the f-th firing sequence associated to the set $T_{q,h}^{im}$, it can be considered made up of the union of $\mathbb{S}_{q,h,f}$ disjoint subsets of concurrent transition firings. Hence, firing of transitions in $T_{q,h}^{im}$ can be considered occurred in $\mathbb{S}_{q,h,f}$ substeps; each substep is denoted q_s, with $s \in [2, \mathbb{S}_{q,h,f}+1]$. Finally, it holds that $T_{q,h}^{im} = \bigcup_{s=2}^{\mathbb{S}_{q,h,f}+1} T_{q_s,h,f}^{im}$.

As indicated in Fig. 5, the marking reached after the firing of transitions in the last of these subsets is a tangible marking corresponding to $\boldsymbol{m}_q$.

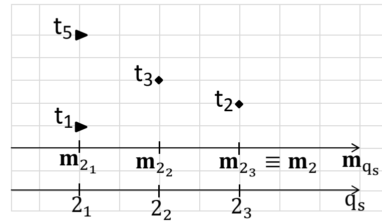

Fig. 5. Possible firing sequence for T_2 of Example 2.

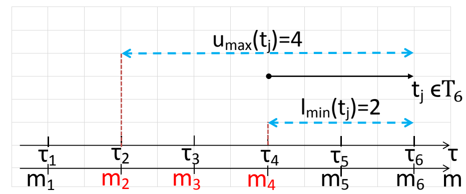

Fig. 6. Set of candidate markings (in red) for the enabling of transition t_j fired at τ_q.

With reference to Example 2, in Fig. 5, a possible solution when $T_2 = \{t_1, t_2, t_3, t_5\}$ is shown. In detail, $T_{2,7}^t = \{t_1, t_5\}$, $T_{2,7}^{im} = \{t_2, t_3\}$; moreover, since it is assumed that t_2 and t_3 fire sequentially according to the second possible firing sequence of the previous example, $T_{2_2,7,2}^{im} = \{t_3\}$, $T_{2_3,7,2}^{im} = \{t_2\}$. In figure, triangles indicate the firing of timed transitions, which leads to the reaching of the first vanishing marking $\boldsymbol{m}_{2_1}$, while diamonds indicate the immediate firings of transitions in $T_{2,7}^{im}$.

The firing of controllable transitions at the q-th step (i.e., $T^c \bigcap T_q \neq \emptyset$) reduces the number of possible couples $(T_{q,h}^t, T_{q,h}^{im})$. As example, with reference to Fig. 4, assuming that t_2 is a controllable transition entails that $t_2 \in T_2^{im}$ and hence the couples $(T_{2,2}^t, T_{2,2}^{im})$; $(T_{2,5}^t, T_{2,5}^{im})$; $(T_{2,8}^t, T_{2,8}^{im})$; $(T_{2,9}^t, T_{2,9}^{im})$; $(T_{2,11}^t, T_{2,11}^{im})$; $(T_{2,12}^t, T_{2,12}^{im})$; $(T_{2,14}^t, T_{2,14}^{im})$; $(T_{2,15}^t, T_{2,15}^{im})$ are inadmissible.

Hence the number of all possible couples reduces to $c_q = \sum_{i=1}^{n_q} \binom{n_q}{i} - \sum_{k=1}^{n_{c,q}} \binom{n_{c,q}}{k} \cdot \sum_{j=0}^{n_q - n_{c,q}} \binom{n_q - n_{c,q}}{j}$ where $n_{c,q} = |T^c \bigcap T_q|$.

Assumption 4: A maximum firing time upper bound $u_{max}(t_j)$, i.e., a time such that $u_j \leq u_{max}(t_j)$, and a minimum firing time lower bound $l_{min}(t_j)$, i.e., a time such that $l_j \geq l_{min}(t_j)$ are available for each timed transition $t_j \in T$.

Given a marking $\boldsymbol{m}_k$, reached at time τ_k, if a timed transition t_j fires later than $\tau_k + u_{max}(t_j)$ then $\boldsymbol{m}_k$ does not enable the firing of t_j. When $u_{max}(t_j)$ is not explicitly defined, then it is assumed $u_{max}(t_j) = \infty$ and consequently firing of t_j at step q can be enabled by any marking $\boldsymbol{m}_k$ reached at time $\tau_k < \tau_q$. Given a marking $\boldsymbol{m}_k$, reached at time τ_k, a transition t_j, enabled by $\boldsymbol{m}_k$ fires in a time greater than or equal to $l_{min}(t_j)$. When $l_{min}(t_j)$ is not explicitly defined, then it is assumed $l_{min}(t_j) = 0$. $\diamond$

As example, in red in Fig. 6 it is shown the set of markings that can enable the firing of t_j at τ_6, assuming $u_{max}(t_j) = 4$ and $l_{min}(t_j) = 2$: marking $\boldsymbol{m}_1$ does not enable the firing of t_j because $\tau_6 - \tau_1 > u_{max}(t_j)$ as well as $\boldsymbol{m}_5$ cannot enable the firing of t_j because $\tau_6 - \tau_5 < l_{mim}(t_j)$.

Assumption 4 reduces to the knowledge of minimum and maximum bounds for the timing structure. This is reasonable in a manufacturing system where an estimate of activities time duration is available, since at least the order of magnitude of such activities can be devised from considerations on their nature (e.g. thermal, electrical, etc). This helps to devise counterexamples, that are sequences that do not belong to the timed language of a TPN.

Note that the approach used in the paper works also without Assumption 4, since the knowledge of minimum and maximum bounds for the timing structure just accelerates the identification procedure by introducing a reduced set of counterexamples.

Definition 5 (Timed Firing Subsequence): Given a timed firing sequence σ_T of length L, $\sigma_{T,q}$ is a subsequence of length q of σ_T if it exists a sequence $\sigma_{T,L-q}$ of length $L-q$ such that $\sigma_T = \sigma_{T,q}\sigma_{T,L-q}$. Subsequence $\sigma_{T,0}$ is equal to $(\epsilon, 0)$, i.e., the firing of empty string at $\tau = 0$. $\diamond$

Definition 6 (Counterexample): Given an observed timed firing sequence σ_T, having length L and a transition $t_j \in T \backslash T_q$ then

$\sigma_{T,q-1}(t_j, \tau_q)$, with $q \in [1, L[$, is a counterexample iff

(i) $\exists k \ : \ \delta(t_j, k, q) = \tau_q - \tau_k \in I(t_j)$ and $\tau_k + u_j \leq \tau_L$

(ii) $\nexists (T_x, \tau_x) \ : \ \tau_x \in [\tau_k + l_j, \tau_k + u_j]$ and $t_j \in T_x$. $\diamond$

In words, $\sigma_{T,q-1}(t_j, \tau_q)$ is a counterexample iff for each step q of the observed timed firing sequence σ_T, for each transition t_j that does not fire at time τ_q: i) there exists at least a step k at which the firing of t_j at τ_q could have been enabled; ii) there does not exist a time instant $\tau_x \in [\tau_k + l_j, \tau_k + u_j]$ when the firing of t_j has been observed.

The computation of all counterexamples requires the knowledge of the timed net language $L(\mathcal{T})$, which is assumed to be not available in this paper. Indeed, the proposed approach is based on a set of observed timed sequences, that are a subset of the timed net language. However, by using $l_{min}(t_j)$ and $u_{max}(t_j)$ in Definition 6 instead of l_j and u_j, which are unknown in the problem considered in this paper, a reduced set of counterexamples is obtained.

As example, consider the net system of Fig. 7(a): assume that $\sigma_T = (\{t_1\}, 1)\ (\{t_2\}, 2)\ (\{t_1\}, 4)\ (\{t_2\}, 6)\ (\{t_1\}, 8)\ (\{t_2\}, 9)$. On the basis of Definition 6, since $I(t_1) = I(t_2) = [1, 2]$, the set of counterexamples is made up of the following

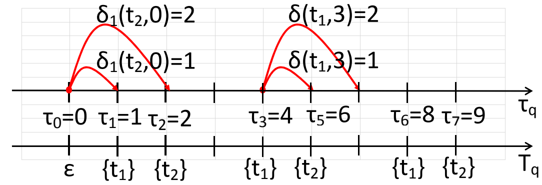

Fig. 7. The sequence $\sigma_{T,3}(\{t_1\}, 6)$ is a counterexample while $\sigma_{T,0}(\{t_2\}, 1)$ is not a counterexample, in view of the fact that condition (ii) is violated, since $\tau_k + l_2 = 1$ and $\tau_k + u_2 = 2$ and $\exists (T_2, \tau_2) \ : \ \tau_2 = 2$ and $t_2 \in T_2$.

timed firing sequences:

$\sigma_{T,4}(\{t_2\}, 8) = (\{t1\}, 1)\ (\{t2\}, 2)\ (\{t1\}, 4)\ (\{t2\}, 6)\ (\{t2\}, 8)$

$\sigma_{T,3}(\{t_1\}, 6) = (\{t_1\}, 1)\ (\{t_2\}, 2)\ (\{t_1\}, 4)\ (\{t_1\}, 6)$

$\sigma_{T,2}(\{t_2\}, 4) = (\{t_1\}, 1)\ (\{t_2\}, 2)\ (\{t_2\}, 4)$.

Consider the second counterexample: condition (i) of Definition 6 is satisfied for $k = 3$ since $\tau_k = 4$; condition (ii) is satisfied too, since $\tau_k + l_1 = 5$ and $\tau_k + u_1 = 6$ and it exists $\tau_5 \in [5, 6]$, but $t_1 \notin T_5$. Instead the sequence $\sigma_{T,0}(\{t_2\}, 1)$ is not a counterexample: condition (i) of Definition 6 is satisfied for $k = 0$ since $\tau_k = 0$; condition (ii) is violated, since $\tau_k + l_2 = 1$ and $\tau_k + u_2 = 2$ and $\exists (T_2, \tau_2) \ : \ \tau_2 = 2$ and $t_2 \in T_2$ (see Fig. 7(b)).

If l_1 (l_2) and u_1 (u_2) are replaced with $l_{min}(t_1) = 0$ ($l_{min}(t_2) = 0$) and $u_{max}(t_1) = 3$ ($u_{max}(t_2) = 3$) no counterexamples are found.

V. Identification of TPN systems

In this section the formal definition of the identification problem is given. In Subsection V-A all the logical conditions that must be satisfied to identify the unknown system are presented and explained, in Subsection V-B the constraints that must be fulfilled to impose that choices involve only controllable transitions are presented and explained, in Subsection V-C the Mixed Integer Linear Programming Problem (MILPP) is presented. A set of rules transforming logical propositions into linear inequalities, representing the enabling and disabling conditions for the firing of transitions included in each observed timed sequence, are presented in V-D, while in Subsection V-E an example of transformation of one MILPP logical condition in a set of linear constraints is given. Finally in Subsection V-G an example to show the effectiveness of the approach is discussed.

The identification problem is formally stated as follows:

Given the observed language $\mathcal{L}_{obs}$ of the unknown system $\mathcal{T}'$ and a set of places P of cardinality m, the problem consists in the identification of the net structure N, the timing structure I, and the initial marking $\boldsymbol{m}_0$ such that the timed language generated by $\mathcal{T} = \langle N, \boldsymbol{m}_0, I \rangle$, named $L(\mathcal{T})$, contains $\mathcal{L}_{obs}$, i.e., $L(\mathcal{T}) \supseteq \mathcal{L}_{obs}$.

The unknowns are the matrices **Pre**, **Post**, *the vector $\boldsymbol{m}_0$ and the firing time lower and upper bounds of each transition $t_j \in T$.*

The proposed approach is based on the formulation of a MILPP where the constraints are obtained by the set of logical conditions, named $\mathcal{G}(\sigma_T)$, representing the enabling and disabling conditions for the firing of transitions included in each observed timed sequence σ_T. These linear inequalities are obtained transforming the logical conditions presented in Subsections V-A according to rules presented in Section V-D and they are linearized as shown in Section V-E. After such a linearization, $\mathcal{G}(\sigma_T)$ provides a linear algebraic characterization of the TPN system with m places and n transitions, such that $L(\mathcal{T}) \supseteq \mathcal{L}_{obs}$. Notice that while n is known, m is unknown. A common approach [22] is to assign to the number of the places a starting value (e.g. $m = \overline{m}$) and try to solve the system of equations: if it gives no solutions, $\overline{m}$ is incremented. On the other hand, if a solution is found, m can be progressively reduced of one place at time to obtain a more compact model until no solution is found. In words, the cardinality of the set P is assumed to be known when $\mathcal{G}(\sigma_T)$ is written. The chosen of the starting value of the number of places comes from heuristic reasoning that is not the focus of this paper. As example, for a sequential process the number of transitions and places is equal. When a system includes concurrent processes, usually the number of places is greater than the number of transitions, at least one additional place for each concurrent process must be considered.

A. Enabling and disabling conditions for transitions

To write in a more compact way the enabling and disabling conditions for transitions, the function $Prev(t_j, q) : T \times \mathbb{N} \rightarrow \mathbb{R}_0^+$ is introduced; it is the function that, given a transition t_j fired at τ_q, returns (if there exists) the time of the last firing of t_j, occurred before of step q, otherwise it returns τ_q. As example, if $\sigma_T = (t_1, 1)(t_2, 3)(t_1, 4)$, $Prev(t_1, 3) = 1$ while $Prev(t_2, 2) = 3$.

The function $\tau_{en}(t_j, k, q) : T \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}_0^+$ is defined as follows: $\tau_{en}(t_j, k, q) = \tau_k$ if $\tau_k \geq Prev(t_j, q)$, otherwise $\tau_{en}(t_j, k, q) = Prev(t_j, q)$. It is a function that, given a transition t_j fired at τ_q, a step k preceding the step q, and the step q, returns the time τ_k if there is not any other firing of t_j occurred at any steps between k and q, otherwise it returns the time of the last firing of t_j. As example, if $\sigma_T = (t_1, 1)(t_2, 3)(t_1, 4)(t_2, 6)$, since $\tau_k = 1$ and $Prev(t_2, 4) = 3$, $\tau_{en}(t_2, 1, 4) = 3$.

Let $r_{k,j,q}$ be a boolean variable equal to 0 iff marking $\boldsymbol{m}_k$ enables the timed firing of t_j at time τ_q.

Proposition 1: Consider the set $T_q = T^t_{q,h} \bigcup T^{im}_{q,h}$ of transitions fired at τ_q (see Section IV), the marking $\boldsymbol{m}_k : \boldsymbol{m}_0[\sigma_{T,k}\rangle \boldsymbol{m}_k$ enables the firing of $t_j \in T^t_{q,h}$ if the following logical condition is satisfied:

$\mathcal{M}^t_e(t_j, q, k)$:

$$\begin{aligned} &\underbrace{\boldsymbol{m}_k \geq \mathbf{Pre}(\cdot, t_j)}_{2.1} \bigwedge \underbrace{(l_j \leq \tau_q - \tau_{en}(t_j, k, q) \leq u_j)}_{2.2} \\ &\bigwedge \underbrace{\sum_{\overline{q}=1}^{q-1} r_{k,j,\overline{q}} \geq 1}_{2.3} \bigwedge \underbrace{\boldsymbol{m}_k \not\geq 2 \cdot \mathbf{Pre}(\cdot, t_j)}_{2.4} \\ &\bigwedge \underbrace{l_{min}(t_j) \leq \tau_q - \tau_k \leq u_{max}(t_j)}_{2.5} \end{aligned} \tag{2}$$

Proof: The proof follows straightforward by the following considerations.

Condition (2.1) imposes that marking $\boldsymbol{m}_k$ enables the firing of $t_j \in T^t_{q,h}$; condition (2.2) imposes that the time elapsed from the enabling of t_j and its firing belongs to its firing interval. Condition (2.3) imposes that just one firing of each transition t_j can be associated to each marking $\boldsymbol{m}_k$; condition (2.4), moreover, imposes each marking $\boldsymbol{m}_k$ cannot enable a number of contemporaneous firings of t_j greater than 1. The conditions (2.3-4) are needed since single server semantic policy has been assumed. Condition (2.5) comes from the knowledge of the maximum upper bound and minimum lower bound of the t_j firing time. ■

From now on it is referred to (2.1) as *state enabling* condition, to (2.2) as *time constraint* condition and to (2.3-4) as *single server semantic* condition.

TPN semantic imposes that an enabled transition t_j must fire in a time belonging to its firing interval, this means that any marking satisfying condition (2) can be selected as responsible of the firing of transitions belonging to $T^t_{q,h}$.

To make a conservative choice with respect to the *a priori* knowledge of the value of the minimum and maximum bound of each $I(t_j)$, if more than one marking $\boldsymbol{m}_k$ satisfies condition (2) then the *oldest* one must be selected as responsible of the firing of transitions belonging to $T^t_{q,h}$. At this aim an additional logical condition must be considered.

Consider the logical predicate

$$LP(\boldsymbol{x}) = \bigwedge_j \boldsymbol{a}_j \cdot \boldsymbol{x} + \boldsymbol{b}_j \leq \boldsymbol{0} \tag{3}$$

where $\boldsymbol{x}$ is a vector variable and $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ are vectors of known coefficients, with the same size of $\boldsymbol{x}$, and consider the ordered set $\{\overline{\boldsymbol{x}_0}, \overline{\boldsymbol{x}_1} \ldots, \overline{\boldsymbol{x}_n}\}$ of values of the vector variable $\boldsymbol{x}$.

The following logical statement imposes that $r_i = 0$ iff $\overline{\boldsymbol{x}_i}$ is the first value that satisfies the predicate (3).

$$\begin{array}{lll} \text{IF} & LP(\boldsymbol{x}_0) & \text{THEN } r_0 = 0 \\ \text{ELSE IF} & LP(\boldsymbol{x}_1) & \text{THEN } r_1 = 0 \\ \vdots & & \\ \text{ELSE IF} & LP(\boldsymbol{x}_n) & \text{THEN } r_n = 0 \end{array} \tag{4}$$

Let h_q be the number of markings that meet condition (2.5). The logical condition used to select the oldest marking $\boldsymbol{m}_k$ that satisfies condition (2) is obtained from (4), replacing in (3) LP by $\mathcal{M}^t_e(t_j, q, k)$, with $n = h_q$, and it is called $\mathcal{M}_{rules}(t_j, q)$.

The logical condition for the enabling of a single transition in $T^t_{q,h}$ can be extended to the whole set of transitions $T^t_{q,h}$. Denote as $\mathcal{M}_{rules}(T^t_{q,h}, q)$ the logical condition $\bigwedge_{\forall t_j \in T^t_{q,h}} \mathcal{M}_{rules}(t_j, q)$.

Proposition 2: Consider the set $T^{im}_{q,h} = \bigcup_{s=2}^{\mathbb{S}_{q,h,f}+1} T^{im}_{q_s,h,f}$.

The marking $\boldsymbol{m}_{q_s} : \boldsymbol{m}_q[T^t_{q,h}\rangle \boldsymbol{m}_{q_1}[\bigcup_{\overline{s}=2}^{s} T^{im}_{q_{\overline{s}},h,f}\rangle \boldsymbol{m}_{q_s}$, where $\boldsymbol{m}_q$ is a reachable marking, enables the firing of each transition $t_j \in T^{im}_{q_s,h,f}$ if the following logical condition is satisfied:

$\mathcal{M}^{im}_e(T^{im}_{q_s,h,f}, q, s, f)$:

$$\underbrace{\boldsymbol{m}_{q_s} \geq \sum_{\forall t_j \in T^{im}_{q_s,h,f}} \mathbf{Pre}(\cdot, t_j)}_{5.1} \tag{5}$$
$$\bigwedge_{\forall t_j \in T^{im}_{q_s,h,f}} \underbrace{l_j = 0}_{5.2}$$
$$\bigwedge_{\forall t_x \in T^{im}_{q,h} \setminus T^{im}_{q_s,h,f}} \underbrace{\boldsymbol{m}_{q_s} < \mathbf{Pre}(\cdot, t_x)}_{5.3}$$

Proof: Condition (5.1) imposes that $\boldsymbol{m}_{q_s}$ enables the firing of all transitions in $T^{im}_{q_s,h,f}$; since all transitions in $T^{im}_{q_s,h,f}$ are enabled by the firing of transitions in $T^t_{q,h}$, they are enabled by a vanishing marking and their lower bound is equal to zero, consequently condition (5.2) holds. Consider a transition $t_x \in T^{im}_{q,h} \setminus T^{im}_{q_s,h,f}$, i.e., a transition that does not fires in q_s, then $\boldsymbol{m}_{q_s}$ does not enable its firing, since if *ad absurdum* marking $\boldsymbol{m}_{q_s}$ enabled the firing of $t_x \in T^{im}_{q,h} \setminus T^{im}_{q_s,h,f}$, as consequence t_x would fire at q_s and hence $t_x \in T^{im}_{q_s,h,f}$. ■

The logical condition for the enabling of concurrent firings of the transitions in the set $T^{im}_{q_s,h,f}$ can be extended to the whole subsets of transitions $T^{im}_{q,h}$. Denote as $\mathcal{M}^{im}_e(T^{im}_{q,h}, q, f)$ the logical condition $\bigwedge_{s=1}^{\mathbb{S}_{q,h,f}} \mathcal{M}^{im}_e(T^{im}_{q_s,h,f}, q, s, f)$.

Proposition 3: For each couple $(T^t_{q,h}, T^{im}_{q,h})$, the set of markings that enables each firing of the transitions in $T^t_{q,h}$ and $T^{im}_{q,h}$, satisfies the logical condition

$\mathcal{M}^h_e(T^t_{q,h}, T^{im}_{q,h})$:

$$\mathcal{M}_{rules}(T^t_{q,h}, q) \bigwedge_{f=1}^{c_{q,h}} \mathcal{M}^{im}_e(T^{im}_{q_s,h,f}, q, f) \tag{6}$$

Proof: Proof follows straightforward from the previous discussion. ■

Let EM_{jq} be the set of tangible markings satisfying condition (2.5) for transition t_j.

Given the h-th couple $(T^t_{q,h}, T^{im}_{q,h})$, let $r_{q,s,j,h,f}$ ($r_{k,j,h}$) be a variable equal to 0 iff: i) $t_j \in T^{im}_{q,h,f}$ ($t_j \in T^t_{q,h}$) and ii) the vanishing marking $\boldsymbol{m}_{q_s}$ (tangible $\boldsymbol{m}_k$) enables its firing; then the following equation can be written:

$$\sum_{h=1}^{c_q} \Big(\sum_{\forall \boldsymbol{m}_k \in EM_{jq}} r_{k,j,h} + \sum_{f=1}^{c_{q,h}} \sum_{s=1}^{\mathbb{S}_{q,h,f}} r_{q,s,j,h,f} \Big) = 1, \ \forall t_j \in T_q \tag{7}$$

Equation (7) assures that the enabling of each timed (immediate) firing of t_j is associated to just one tangible (vanishing) marking.

Proposition 4: The set of tangible and vanishing markings that enable each firing of transitions in T_q, satisfies the logical condition

$\mathcal{M}_e(T_q, q)$:

$$\bigwedge_{h=1}^{c_q} \mathcal{M}^h_e(T^t_{q,h}, T^{im}_{q,h}, h) \bigwedge$$
$$\sum_{h=1}^{c_q} \big(\sum_{\forall \boldsymbol{m}_k \in EM_{jq}} r_{k,j,h} + \sum_{f=1}^{c_{q,h}} \sum_{s=1}^{\mathbb{S}_{q,h,f}} r_{q,s,j,h,f} \big) = 1, \quad \forall t_j \in T_q \tag{8}$$

Proof: Proof follows straightforward from the previous discussion. ■

If a marking $\boldsymbol{m}_k$ does not satisfy (2.5), it cannot enable the firing of t_j at step q. This allows to devise an additional set of linear constraints as shown in Proposition 5.

Proposition 5: Consider the transition $t_j \in T_q$ fired at step q. Given a marking $\boldsymbol{m}_k$ whose indexes satisfy conditions $k < q$, $\tau_k + u_{max}(t_j) < \tau_q$, $\tau_k > Prev(t_j, q)$ then the following condition holds

$\mathcal{M}_d(t_j, q, k)$:

$$\boldsymbol{m}_k < \mathbf{Pre}(\cdot, t_j) \tag{9}$$

Proof: A marking $\boldsymbol{m}_k$ reached at time τ_k such that $\tau_k + u_{max}(t_j) < \tau_q$ and $\tau_k > Prev(t_j, q)$ cannot enable the firing of t_j at time τ_q since a transition surely fires if a time equal to u_j is elapsed from its enabling.

This means that $\boldsymbol{m}_k < \mathbf{Pre}(\cdot, t_j)$, i.e., condition (2.1) is violated. ■

Proposition 6 is used to obtain the constraints that characterize the counterexamples of the observed timed sequence.

Proposition 6 (Counterexample system): Consider a counterexample $\sigma'_T = \sigma_{T,q-1}(t_j, \tau_q)$ such that $\sigma'_T \notin \mathcal{L}_{obs}$ and a marking $\boldsymbol{m}_k$ whose indexes satisfy conditions $k < q$, $l_{min}(t_j) \leq \tau_q - \tau_k \leq u_{max}(t_j)$. The following condition must hold

$\mathcal{M}_c(t_j, q, k)$:

$$\boldsymbol{m}_k < \mathbf{Pre}(\cdot, t_j) \bigvee \tau_q - \tau_{en}(t_j, k, q) \notin [l_j, u_j] \tag{10}$$

Proof: Proof follows from definition of counterexample given in Proposition 6. ■

B. Controllable Choices Conditions

The following lemma, that is a slightly modified version of the one presented in [9], introduces a set of linear algebraic constraints that must be fulfilled in order to guarantee that, according to point 2) of Assumption 2, all transitions that make up a choice are controllable.

Lemma 1: Consider a net $N = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, the set $\{t \in T \mid t \in p^\bullet, \ |p^\bullet| > 1\}$, i.e., the set of transitions involved in a choice, is composed by only controllable transitions if and only if the following set of linear algebraic constraints is fulfilled

$$\mathcal{B}(p_i): \begin{cases} \boldsymbol{e}_i^T \cdot \mathbf{Pre} \cdot \boldsymbol{e}_j - z^i_j V \leq 0 \quad \forall \ t_j \in T & \text{(11a)} \\ \sum_{\forall \alpha | t_\alpha \in T^{uc}} z^i_\alpha \leq 1 & \text{(11b)} \\ (n - n_{uc}) \cdot \sum_{\forall \alpha | t_\alpha \in T^{uc}} z^i_\alpha + \sum_{\forall \beta | t_\beta \in T^c} z^i_\beta \leq n - n_{uc} & \text{(11c)} \\ \mathbf{Pre} \in \mathbb{N}^{m \times n} & \text{(11d)} \\ z^i_j \in \mathbb{N} & \text{(11e)} \end{cases}$$

for all places $p_i \in P$. The constant V is such that $V > \max_{i,j} \boldsymbol{e}_i^T \cdot \mathbf{Pre} \cdot \boldsymbol{e}_j$ where $\boldsymbol{e}_i$ is the i-th canonical basis vector.

Proof: *(if)*. Let p_i be a place such that $|p_i^\bullet| > 1$. Suppose, *ad absurdum*, that constraints (11) are fulfilled but there exists an uncontrolled transition $t_j \in T^{uc}$, such that $t_j \in p_i^\bullet$. In order to fulfill (11a) it should be $z^i_j = 1$. Since $|p_i^\bullet| > 1$, there should exist at least another transition $t_l \in T$, with $l \neq j$, such that $z^i_l = 1$ in order to fulfill the constraint (11a). It readily follows that condition (11c) cannot be fulfilled anymore, which contradicts the initial hypothesis.

(only if).Given a place p_i, let suppose that constraints (11) are not fulfilled and, *ad absurdum*, that the assumption that all transitions that make up a choice are controllable holds. It is

straightforward to show that if just one of the constraints (11a)-(11c) is not fulfilled, then it is $|p_i^\bullet| > 1$, hence assumption cannot hold, contradicting the initial hypothesis. ■

Constraint (11a), when $z_j^i = 0$, imposes that $\mathbf{Pre}(p_i, t_j) \leq 0$, i.e., there does not exist an arc that starting from p_i enters in t_j, while it is redundant for $z_j^i > 0$; constraint (11b) imposes that each place p_i belongs to the preset of no more than one uncontrollable transition, i.e., there does not exist a choice involving two or more uncontrollable transitions; constraint (11c) imposes that there does not exist a choice involving both controllable and uncontrollable transitions.

C. MILPP formulation of identification problem

For the sake of brevity, from now on, $\mathcal{G}(\sigma_T)$, $\mathcal{M}_e(T_q, q)$, $\mathcal{M}_d(t_j, q, k)$ and $\mathcal{M}_c(t_j, q, k)$ denote both the logical conditions and the set of linear constraints obtained applying the rules presented in Section V-D.

The following set of equations, named $\mathcal{C}(T_q, \tau_q)$ is obtained converting the logical conditions devised in Subsections V-A into linear constraints:

$$\mathcal{C}(T_q,\tau_q):\begin{cases} \mathcal{M}_e(T_q, q) & \text{(12a)}\\ \mathcal{M}_d(t_j, q, k),\ \forall t_j \in T_q & \text{(12b)}\\ \quad \forall k\ :\ \tau_k + u_{max}(t_j) < \tau_q & \\ \mathcal{M}_c(t_j, q, k),\ \forall t_j \in T \setminus T_q, & \text{(12c)}\\ \quad \text{s.t. } \sigma_{T,q-1}(t_j, \tau_q) \text{ is a counterexample} & \\ \quad \forall k\ :\ l_{min}(t_j) \leq \tau_q - \tau_k \leq u_{max}(t_j) & \end{cases}$$

Let $\mathcal{L}_{obs}$ be the observed language of the unknown system. A solution to the identification problem can be computed solving the system of equations $\bigcap_{\sigma_T \in \mathcal{L}_{obs}} \mathcal{G}(\sigma_T)$, where

$$\mathcal{G}(\sigma_T):\begin{cases} \mathcal{C}(T_q, \tau_q),\ \forall\ (T_q, \tau_q) \in \sigma_T & \text{(13a)}\\ u_j - l_j \geq 0,\ \forall\ t_j \in T & \text{(13b)}\\ u_j = 0,\ \forall\ t_j \in T^c & \text{(13c)}\\ l_j = 0,\ \forall\ t_j \in T^c & \text{(13d)}\\ \boldsymbol{m}_k \geq \mathbf{0},\ \forall\ k \leq L & \text{(13e)} \end{cases}$$

Equation (13b) imposes that, for each $t_j \in T$, value of u_j (the t_j maximal firing time) is greater than or equal to l_j (the t_j minimum firing time); equations (13c-13d) impose $I(t_j) = [0, 0]$ for each controllable transitions; equation (13e) imposes that each marking $\boldsymbol{m}_k$, reached by the identified net, is a feasible marking.

In general the solution of the $\mathcal{G}(\sigma_T)$ is not unique, thus there exists more than one TPN system $\mathcal{T}$ such that $L(\mathcal{T}) \supseteq \mathcal{L}_{obs}$. To select one among these systems a performance index is given and, solving an appropriate MILPP, a TPN system that minimizes the considered performance index is determined. In particular, if $f(\boldsymbol{m}_0, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{l}, \boldsymbol{u})$ is the considered performance index, where $\boldsymbol{l}, \boldsymbol{u} \in \mathcal{Q}^n$ are, respectively, the vectors of the firing times lower and upper bounds, given the place set P of the system to identify, an identification problem can be formally stated as follows

$$\min_{s.t. \mathcal{G}(\sigma_T) \forall \sigma_T \in \mathcal{L}_{obs},\, B(p_i) \forall p_i \in P} f(\boldsymbol{m}_0, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{l}, \boldsymbol{u}) \tag{14}$$

where $B(p_i)$ has been defined in Lemma 1.

Different choices can be made for the cost function, in particular if the cost function is chosen as

$$\begin{aligned} &f(\boldsymbol{m}_0, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{l}, \boldsymbol{u}) = \\ &\mathbf{1_m^T} \cdot \boldsymbol{m_0} + \mathbf{1_m^T} \cdot (\mathbf{Pre} + \mathbf{Post}) \cdot \mathbf{1_n} - \mathbf{1_n^T} \cdot \mathbf{l} + \mathbf{1_n^T} \cdot \mathbf{u}\,, \end{aligned} \tag{15}$$

the solution minimizes the sum of the tokens in the initial marking, the sum of the arc weights [3] and the width of the firing interval $I(t_j)$ for each transition.

D. Transformation of logical propositions into linear inequalities

In the following a set of rules transforming logical propositions, devised in Section V, into linear inequalities are presented. Two rules from [3] are first recalled, then new transformations are presented.

R 1: **Inequality constraints.** Consider the constraint $\bigvee_{i=1}^{r} \boldsymbol{a}_i \leq \mathbf{0_n}$ where $\boldsymbol{a}_i \in \mathbb{R}^n$, $i = 1, \ldots, r$, and $\bigvee$ denotes the logical or operator. Such constraint can be rewritten as linear algebraic constraints:

$$\begin{cases} \boldsymbol{a}_1 \leq z_1 \cdot \boldsymbol{K} \\ \vdots \\ \boldsymbol{a}_r \leq z_r \cdot \boldsymbol{K} \\ z_1 + \cdots + z_r = r - 1 \\ z_1, \ldots, z_r = \{0, 1\} \end{cases} \tag{16}$$

where $\mathbf{K}$ is any constant vector in $\mathbb{R}^n$ that satisfies the following relation:

$K_j > \max_{i \in \{1,\ldots,r\}} a_i(j), \quad j = 1, \ldots, n$ ◊

R 2: **Equality constraints.** Consider the constraint $\bigvee_{i=1}^{r} \boldsymbol{a}_i = \boldsymbol{b}_i$ where $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{R}^n$, $i = 1, \ldots, r$. Such constraint can be rewritten as linear algebraic constraints:

$$\begin{cases} \boldsymbol{a}_1 - \boldsymbol{b}_1 \leq z_1 \cdot \boldsymbol{K} \\ \boldsymbol{a}_1 - \boldsymbol{b}_1 \geq -z_1 \cdot \boldsymbol{K} \\ \vdots \\ \boldsymbol{a}_r - \boldsymbol{b}_r \leq z_r \cdot \boldsymbol{K} \\ \boldsymbol{a}_r - \boldsymbol{b}_r \geq -z_r \cdot \boldsymbol{K} \\ z_1 + \cdots + z_r = r - 1 \\ z_1, \ldots, z_r = \{0, 1\} \end{cases} \tag{17}$$

where $\boldsymbol{K}$ is any constant vector in $\mathbb{R}^n$ that satisfies the following relation:

$K_j > \max_{i \in \{1,\ldots,r\}} |a_i(j) - b_i(j)|, \quad j = 1, \ldots, n$ ◊

Inspired by the results in [36] and [3], some other rules to convert logical propositions into linear inequalities are here presented.

R 3: The logical propositions

$$\boldsymbol{a} \geq \boldsymbol{b}\ (\boldsymbol{a} < \boldsymbol{b}) \leftrightarrow z = 0;\ (\overline{z} = 0)$$

where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ and $\leftrightarrow$ stands for "*if and only if*", can be rewritten in terms of algebraic constraints as:

$$\begin{cases} \boldsymbol{a} + z \cdot \boldsymbol{K} \geq \boldsymbol{b} & \text{(18a)}\\ \boldsymbol{a} - \overline{z} \cdot \boldsymbol{K} < \boldsymbol{b} & \text{(18b)}\\ z + \overline{z} = 1 & \text{(18c)}\\ z, \overline{z} \in \{0, 1\} & \text{(18d)}\\ \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n & \text{(18e)} \end{cases}$$

with $\boldsymbol{K} \in \mathbb{R}^n\ :\ K_j > |\boldsymbol{a}(j) - \boldsymbol{b}(j)|$ ◊

Proof: *(if)* Suppose $\boldsymbol{a} \geq \boldsymbol{b}$ and *ad absurdum* $z = 1$. Eq.(18c) imposes $\overline{z} = 0$ consequently Eq. (18a) is satisfied for each value of $\boldsymbol{a}$ but Eq. (18b) is violated;

(only if) suppose now that $z = 0$ and *ad absurdum* $\boldsymbol{a} < \boldsymbol{b}$: Eq.(18a) is violated. ■

R 4: The logical proposition

$$\boldsymbol{a} = \boldsymbol{b} \ (\boldsymbol{a} \neq \boldsymbol{b}) \leftrightarrow z = 0 \ (\overline{z} = 0)$$

where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ and can be rewritten in terms of algebraic constraints as:

$$\begin{cases} \boldsymbol{a} + z \cdot \boldsymbol{K} \geq \boldsymbol{b} & (19a) \\ \boldsymbol{a} - z \cdot \boldsymbol{K} \leq \boldsymbol{b} & (19b) \\ \boldsymbol{a} - \overline{z} \cdot \boldsymbol{K} + z_g \cdot \boldsymbol{K} > \boldsymbol{b} & (19c) \\ \boldsymbol{a} - \overline{z} \cdot \boldsymbol{K} - z_l \cdot \boldsymbol{K} < \boldsymbol{b} & (19d) \\ z + \overline{z} = 1 & (19e) \\ z_g + z_l = 1 & (19f) \\ z, \overline{z}, z_g, z_l \in \{0, 1\} & (19g) \\ \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n & (19h) \end{cases}$$

with $\boldsymbol{K} \in \mathbb{R}^n \ : \ K_j > |\boldsymbol{a}(j) - \boldsymbol{b}(j)|$ ◊

Proof: (if) Suppose $\boldsymbol{a} = \boldsymbol{b}$ and *ad absurdum* $z = 1$. Eq. (19a) and (19b) are satisfied for each value of $\boldsymbol{a}$; Eq.(19e) imposes $\overline{z} = 0$ consequently, to satisfy Eq.(19c) that imposes $\boldsymbol{a} > \boldsymbol{b}$, it is necessary that $z_g = 1$ and, at the same way, to satisfy Eq.(19d), that imposes that $\boldsymbol{a} < \boldsymbol{b}$, it must be $z_l = 1$ but in this way Eq.(19f) is violated;

(only if) Suppose $z = 0$ and *ad absurdum* $\boldsymbol{a} \neq \boldsymbol{b}$: it means that either $\boldsymbol{a} > \boldsymbol{b}$ or $\boldsymbol{a} < \boldsymbol{b}$ but if $\boldsymbol{a} > \boldsymbol{b}$, Eq.(19b) is violated while if $\boldsymbol{a} < \boldsymbol{b}$, Eq. (19a) is violated. ■

R 5: The logical proposition

$$\text{IF } \boldsymbol{a} = \boldsymbol{b} \text{ THEN } \boldsymbol{c} \geq \boldsymbol{d} \text{ ELSE } \boldsymbol{e} \leq \boldsymbol{f}$$

where $\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{e}, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{f} \in \mathbb{R}^n$, can be rewritten in terms of algebraic constraints as:

$$\begin{cases} \boldsymbol{a} + z \cdot \boldsymbol{K} \geq \boldsymbol{b} & (20a) \\ \boldsymbol{a} - z \cdot \boldsymbol{K} \leq \boldsymbol{b} & (20b) \\ \boldsymbol{a} + \overline{z} \cdot \boldsymbol{K} + z_g \cdot \boldsymbol{K} > \boldsymbol{b} & (20c) \\ \boldsymbol{a} - \overline{z} \cdot \boldsymbol{K} - z_l \cdot \boldsymbol{K} < \boldsymbol{b} & (20d) \\ \boldsymbol{c} + z \cdot \boldsymbol{K} \geq \boldsymbol{d} & (20e) \\ \boldsymbol{e} - \overline{z} \cdot \boldsymbol{K} \leq \boldsymbol{f} & (20f) \\ z + \overline{z} = 1 & (20g) \\ z_l + z_g = 1 & (20h) \\ z, z_l, z_g \in \{0, 1\} & (20i) \end{cases}$$

with $\boldsymbol{K} \in \mathbb{R}^n \ : \ K_j > \max(|\boldsymbol{a}(j) - \boldsymbol{b}(j)|, |\boldsymbol{c}(j) - \boldsymbol{d}(j)|, |\boldsymbol{e}(j) - \boldsymbol{f}(j)|)$ ◊

Proof: (then) Suppose $\boldsymbol{a} = \boldsymbol{b}$, consequently $z = 0$ (see R4). As consequence of Eq.(20g), $\overline{z} = 1$ and Eq.(20c) and (20d) are satisfied for each value of $\boldsymbol{a}, z_l$ and z_g. To satisfy (20h) it can be chosen $z_l = 0$ and $z_g = 1$ or *vice versa*. Eq. (20f) is satisfied for each value of $\boldsymbol{e}$ and finally Eq.(20e) imposes $\boldsymbol{c} \geq \boldsymbol{d}$.

(else) Suppose now that $\boldsymbol{a} \neq \boldsymbol{b}$ and consequently $z = 1$ and $\overline{z} = 0$. If $\boldsymbol{a} > \boldsymbol{b}$ Eq. (20c) and (20d) are satisfied with $z_g = 0$ and $z_l = 1$, otherwise they hold on if $z_g = 1$ and $z_l = 0$. Eq. (20a) and (20b) are satisfied for each value of $\boldsymbol{a}$ as well as Eq. (20e) holds on for each value of $\boldsymbol{b}$, while Eq. (20f) imposes $\boldsymbol{e} \leq \boldsymbol{f}$. ■

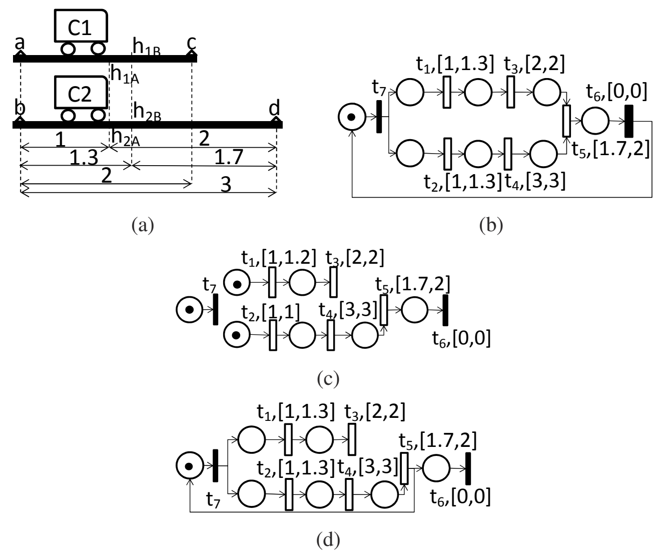

Fig. 8. (a) System of the example: two cars going towards right and returning; (b) TPN system modeling the system; (c) identified system with $\mathcal{L}_{obs} = \{\sigma_T, \sigma'_T\}$; (d) identified system with $\mathcal{L}_{obs} = \{\sigma_T, \sigma'_T, \sigma''_T\}$.

E. Transformation of the logical conditions into linear constraints

Applying the rules introduced in Section V-D, the logical conditions in the set $\mathcal{G}(\sigma_T)$ can be rewritten as sets of linear constraints.

As example the set of constraints equivalent to logical condition $\mathcal{M}_e^t(t_j, q, k)$ is shown, but the same procedure can be applied to conditions $\mathcal{M}_{rules}(T_{q,h}^t, q)$, $\mathcal{M}_e^{im}(T_{q,h}^{im}, q, f)$, $\mathcal{M}_e^h(T_{q,h}^t, T_{q,h}^{im})$, $\mathcal{M}_e(T_q, q)$, $\mathcal{M}_d(t_j, q, k)$, $\mathcal{M}_c(t_j, q, k)$, to obtain the corresponding set of linear constraints.

Given the timed sequence $\sigma_{T,k}$, $\boldsymbol{\sigma}_{T,k} \ : \ T \to \mathbb{N}$ is its firing count vector, where $\boldsymbol{\sigma}_{T,k}(t)$ represents the number of occurrence of t in $\sigma_{T,k}$.

Proposition 7: $\mathcal{M}_e^t(t_j, q, k)$ is equivalent to the following set of linear constraints

$$
\left\{
\begin{array}{l}
\left.\begin{array}{l}
\boldsymbol{m}_0 + \mathbf{Post}\cdot\boldsymbol{\sigma}_{T,k} - \mathbf{Pre}\cdot(\boldsymbol{\sigma}_{T,k}+\boldsymbol{e}_j) + \\
\qquad + z_{k,j,q}\cdot\boldsymbol{K}_1 + \overline{z}_{s_{k,j,q}}\cdot\boldsymbol{K}_1 \geq \mathbf{0_m} \\
\boldsymbol{m}_0 + \mathbf{Post}\cdot\boldsymbol{\sigma}_{T,k} - \mathbf{Pre}\cdot(\boldsymbol{\sigma}_{T,k}-\boldsymbol{e}_j) + \\
\qquad - \overline{z}_{k,j,q}\cdot\boldsymbol{K}_1 - \overline{z}_{s_{k,j,q}}\cdot\boldsymbol{K}_1 \lneq -\mathbf{1_m} \\
z_{k,j,q} + \overline{z}_{k,j,q} = 1 \\
z_{k,j,q}, \overline{z}_{k,j,q} \in \{0,1\}
\end{array}\right\} (21.1) \\
\left.\begin{array}{l}
u_j + z_{k,j,q}\cdot K_2 + d_{k,j,q}\cdot K_2 \geq \tau_q - \tau_{en}(t_j,k,q) \\
l_j - z_{k,j,q}\cdot K_2 - d_{k,j,q}\cdot K_2 \leq \tau_q - \tau_{en}(t_j,k,q) \\
l_j + z_{k,j,q}\cdot K_2 + \overline{d}_{k,j,q}\cdot K_2 + \\
\qquad + z_{g_{ks\underline{jq}}}\cdot K_2 > \tau_q - \tau_{en}(t_j,k,q) \\
u_j - z_{k,j,q}\cdot K_2 - \overline{d}_{k,j,q}\cdot K_2 + \\
\qquad - z_{l_{k,j,q}}\cdot K_2 < \tau_q - \tau_{en}(t_j,k,q) \\
d_{k,j,q} + \overline{d}_{k,j,q} = 1 \\
z_{g_{k,j,q}} + z_{l_{k,j,q}} = 1 \\
d_{k,j,q}, \overline{d}_{k,j,q}, z_{g_{k,j,q}}, z_{l_{k,j,q}} \in \{0,1\}
\end{array}\right\} (21.2) \\
\left.\begin{array}{l}
\sum_{\overline{q}=1}^{q-1} r_{k,j,\overline{q}} + z_{s_{k,j,q}}\cdot K_4 \geq 1 \\
\sum_{\overline{q}=1}^{q-1} r_{k,j,\overline{q}} - \overline{z}_{s_{k,j,q}}\cdot K_4 \leq 0 \\
r_{k,j,q} + z_{s_{k,j,q}}\cdot K_5 \geq 1 \\
r_{k,j,q} - z_{s_{k,j,q}}\cdot K_5 \leq 1 \\
z_{s_{k,j,q}} + \overline{z}_{s_{k,j,q}} = 1 \\
z_{s_{k,j,q}}, \overline{z}_{s_{k,j,q}}, r_{k,j,q} \in \{0,1\}
\end{array}\right\} (21.3) \\
\left.\begin{array}{l}
\boldsymbol{m}_0 + \mathbf{Post}\cdot\boldsymbol{\sigma}_{T,k} - \mathbf{Pre}\cdot(\boldsymbol{\sigma}_{T,k}+2\boldsymbol{e}_j) + \\
\qquad + \overline{z}_{k,j,q}\cdot\boldsymbol{K}_1 + r_{k,j,q}\cdot\boldsymbol{K}_1 \geq \mathbf{0_m} \\
\boldsymbol{m}_0 + \mathbf{Post}\cdot\boldsymbol{\sigma}_{T,k} - \mathbf{Pre}\cdot(\boldsymbol{\sigma}_{T,k}-2\boldsymbol{e}_j) + \\
\qquad - z_{k,j,q}\cdot\boldsymbol{K}_1 - r_{k,j,q}\cdot\boldsymbol{K}_1 \lneq -\mathbf{1_m}
\end{array}\right\} (21.4)
\end{array}
\right. \tag{21}
$$

Proof: From R3 it follows that equations (21.1) implement condition (2.1), equations (21.2) implement condition (2.2), equations (21.3) implement condition (2.3) and (21.4) implement condition (2.4).

In equations (21.1) and (21.4), $\boldsymbol{K}_1 \in \mathbb{R}^m$ and it is such that $\boldsymbol{K}_1 > |\boldsymbol{m}_0 + \mathbf{Post}\cdot\boldsymbol{\sigma}_{T,k} - \mathbf{Pre}\cdot(\boldsymbol{\sigma}_{T,k}+\boldsymbol{e}_j)|$; in equations (21.2), $K_2 \in \mathbb{R}$ and $K_2 > |u_j - (\tau_q - \tau_{en}(t_j,k,q))|$; in equations (21.3), $K_4, K_5 \in \mathbb{N}$ and $K_4, K_5 > 1$. ■

Since all its terms are known, condition (2.5) is not converted into linear constraints but it is used to restrict the set of the indexes k used to individuate the markings $\boldsymbol{m}_k$ to test.

F. Identification Algorithm

In Fig. 9 the overall identification algorithm is shown.

The algorithm is made up of three steps. In the first one, the observation of a new timed firing sequence σ_T is considered. Step 2 starts when it is verified that σ_T does not belong to the language of the current model $\mathcal{T}$ of the system, i.e., the current TPN cannot generate σ_T. In this step the set of logical conditions associated to σ_T are built, then they are transformed into the corresponding linear constraints $\mathcal{G}(\sigma_T)$. After that, Step 3 (building and resolution of the MILPP) can be executed: a new model of the system is obtained and Step 1 can be repeated.

The algorithm considers the case in which the identification of the model is executed soon after an observation, but it can be easily modified to adapt it to the case in which observation and identification are executed separately (as stated at the end of Section III), according to a batch processing approach.

Moreover, the algorithm consider the possibility that a system can be reset. In this case, the algorithm restarts from Step 0, $\sigma_T = \epsilon$ but $\mathcal{L}_{obs}$ contains the whole set of previous observed timed sequences. Then, the MILPP is solved on the basis of all the observed timed sequences.

$\mathcal{L}_{obs} := \emptyset$
Step 0: Initialization of a new firing sequence
$q := 1$
$\sigma_{T,q-1} := \varepsilon$
Step 1: Observation of a new firing sequence
observe a new couple (T_q, τ_q)
$\sigma_T := \sigma_{T,q-1}(T_q, \tau_q)$
The new couple is appended to the current timed firing sequence
q:=q+1
$\mathcal{L}_{obs} := \mathcal{L}_{obs} \bigcup \sigma_T$
The new timed firing sequence is added to the observed language
Step 2: Building of the logical conditions associated to the sequence σ_T
Step 2.a: Logical conditions about transition firings
build all the possible combinations $(T^t_{q,h}, T^{im}_{q,h})$
for each h-th combination $(T^t_{q,h}, T^{im}_{q,h})$
write the logical condition $\mathcal{M}_{rules}(T^t_{q,h}, q)$
build all the possible firing sequences that can be obtained from $T^{im}_{q,h}$
for each f-th firing sequence write the logical condition $\mathcal{M}^{im}_e(T^{im}_{q,h}, q, f)$
write the logical condition $\mathcal{M}^h_e(T^t_{q,h}, T^{im}_{q,h})$
for each $t_j \in T^t_{q,h}$
for k:=0 to q-1
if $l_{min}(t_j) \leq \tau_q - \tau_k \leq u_{max}(t_j)$
$EM_{jq} := \boldsymbol{m}_k$
if $\tau_k \geq Prev(t_j, q)$, $\tau_{en}(t_j, k, q) = \tau_k$
else $\tau_{en}(t_j, k, q) = Prev(t_j, q)$
end
end
The set EM_{jq} of each timed firing has been built
end
write the logical condition $\mathcal{M}_e(T_q, q)$
for each $t_j \in T_q$
for k=1 to q
if $\tau_k + u_{max}(t_j) < \tau_q$ and $\tau_k > Prev(t_j, q)$,
write the logical condition $\mathcal{M}_d(t_j, q, k)$
end
end
Step 2.b: Computation of the counterexamples
for each $t_j \in T \setminus T_q$
if $\exists k\ :\ \boldsymbol{m}_k = \boldsymbol{m}_0[\sigma_{T,k}\rangle\boldsymbol{m}_k$, $\delta(t_j, k, q) = \tau_q - \tau_k \in I(t_j)$
v and $\tau_k + u_j \leq \tau_L$
if $\nexists(T_x, \tau_x)\ :\ \tau_x \in [\tau_k + l_{min}(t_j), \tau_k + u_{max}(t_j)]$ and $t_j \in T_x$
$CE := CE \bigcup \sigma_{T,q-1}(t_j, \tau_q)$
end
end
end
**If $\sigma_{T,q-1}(t_j, \tau_q)$ satisfies all conditions of Definition 6,*
*it is added to the set of counterexamples CE**
Step 2.c: Logical conditions about counterexamples
for each $\sigma_{q-1}(t_j, \tau_q) \in CE$ write the logical condition $\mathcal{M}_c(t_j, q, k)$
Step 2.d: Building of the system $\mathcal{G}(\sigma_T)$
for each place $p_i \in P$ write the logical conditions $B(p_i)$
for q:=1 to $|\sigma_T|$ write linear constraints $\mathcal{C}(T_q, \tau_q)$
write $\mathcal{G}(\sigma_T)$
if $\mathcal{T}$ is the current identified model of the system and $\sigma_T \in \mathcal{L}(\mathcal{T})$, goto Step 4
**When the observed sequence belongs to the language of the current*
*identified model a new observation starts, otherwise the MILPP is resolved**
Step 3: Resolution of the MILPP
write the optimization function $f(\boldsymbol{m}_0, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{l}, \boldsymbol{u})$
build the MILPP (14)
solve the MILPP (14)
Step 4: Start of a new observation
if the system has been reset goto Step 0
goto Step 1

Fig. 9. Identification algorithm

G. Example

The considered example is an adapted version of the system used in [22]. It is made up of two cars C1 and C2 (Fig. 8(a)), that start from an arbitrary position in the home space (delimited by points h_{1A} and h_{1B} for C1 and h_{2A} and h_{2B} for C2, in the figure) and move independently to reach points a and b respectively. When C1 (C2) arrives at a (b), the car starts to move along right direction until c (d) is reached (the time units (t.u.) a car takes to arrive in the designed points are shown in the figure). Then, C1 (C2) stops and remains in this state until both cars are in their right positions. It takes from 1.7 to 2 t.u. to return cars in home position, then a new cycle is

TABLE I

MEANING OF TRANSITIONS OF THE EXAMPLE.

Transition controllable	Transition uncontrollable	Event
	t_1 (t_2)	C1 (C2) arrives at a (b).
	t_3 (t_4)	C1 (C2) arrives at c (d).
	t_5	Both cars have returned in their home position.
	t_6	Both cars are ready to start a new cycle.
t_7		Cycle starts again.

immediately started.

The net modeling such a system is shown in Fig. 8(b); in Table I the meaning of each transition is reported.

Cplex© has been used as mathematical programming solver.

The choice of the starting value of the number of places, as said in Section V, comes from a heuristic reasoning that is not the focus of the paper. In a completely sequential process, a good practice can be to start with m equal to the number of the events. This number should be increased when the process includes concurrent activities, since some places may be required to synchronize them. The proposed example consists in two cars that move concurrently and 7 events. Then, the number of places has been initially set to 9 and then reduced to 7. For smaller values of m, no solution has been found to the MILPP.

The solution of the identification problem (14) formulated from two observed timed sequences, $\sigma_T = (\{t_7\},0)$ $(\{t_1,t_2\},1)$ $(\{t_3\},3)$ $(\{t_4\},4)$ $(\{t_5,t_6\},6)$ (of length $L_1 = 5$) and $\sigma'_T = (\{t_7\},0)$ $(\{t_2\},1)$ $(\{t_1\},1.2)$ $(\{t_3\},3.2)$ $(\{t_4\},4)$ $(\{t_5,t_6\},5.7)$ (of length $L_2 = 6$), when the objective function is the one in (15) and $m = 7$, leads to the PN model represented in Fig. 8(c).

The following values for the minimum lower bound and the maximum upper bound have been fixed: $l_{min}(t_1) = l_{min}(t_2) = l_{min}(t_6) = 1$, $l_{min}(t_3) = l_{min}(t_4) = l_{min}(t_5) = 0$, $u_{max}(t_j) = 3\ \forall t_j$ of the system.

The algorithm autonomously builds the following set of 46 counterexamples:

$\sigma_{T,0}(\{t_1\},0)$, $\sigma_{T,4}(\{t_1\},4)$, $\sigma_{T,5}(\{t_1\},6)$, $\sigma_{T,0}(\{t_2\},0)$, $\sigma_{T,4}(\{t_2\},4)$, $\sigma_{T,5}(\{t_2\},6)$, $\sigma_{T,5}(\{t_3\},6)$, $\sigma_{T,0}(\{t_4\},0)$, $\sigma_{T,1}(\{t_4\},1)$, $\sigma_{T,0}(\{t_5\},0)$, $\sigma_{T,1}(\{t_5\},1)$, $\sigma_{T,3}(\{t_5\},3)$, $\sigma_{T,0}(\{t_6\},0)$, $\sigma_{T,1}(\{t_6\},1)$, $\sigma_{T,3}(\{t_6\},3)$, $\sigma_{T,1}(\{t_7\},1)$, $\sigma_{T,3}(\{t_7\},3)$, $\sigma_{T,4}(\{t_7\},4)$, $\sigma_{T,5}(\{t_7\},6)$, $\sigma'_{T,0}(\{t_1\},0)$, $\sigma'_{T,3}(\{t_1\},3.2)$, $\sigma'_{T,4}(\{t_1\},4)$, $\sigma'_{T,4}(\{t_1\},5.7)$, $\sigma'_{T,0}(\{t_2\},0)$, $\sigma'_{T,3}(\{t_2\},3.2)$, $\sigma'_{T,4}(\{t_2\},4)$, $\sigma'_{T,4}(\{t_2\},5.7)$, $\sigma'_{T,0}(\{t_3\},0)$, $\sigma'_{T,1}(\{t_3\},1)$, $\sigma'_{T,4}(\{t_3\},5.7)$, $\sigma'_{T,0}(\{t_4\},0)$, $\sigma'_{T,1}(\{t_4\},1)$, $\sigma'_{T,0}(\{t_5\},0)$, $\sigma'_{T,1}(\{t_5\},1)$, $\sigma'_{T,2}(\{t_5\},1.2)$, $\sigma'_{T,3}(\{t_5\},3.2)$, $\sigma'_{T,4}(\{t_5\},4)$, $\sigma'_{T,0}(\{t_6\},0)$, $\sigma'_{T,1}(\{t_6\},1)$, $\sigma'_{T,2}(\{t_6\},1.2)$, $\sigma'_{T,3}(\{t_6\},3.2)$, $\sigma'_{T,4}(\{t_6\},4)$, $\sigma'_{T,2}(\{t_7\},1.2)$, $\sigma'_{T,3}(\{t_7\},3.2)$, $\sigma'_{T,4}(\{t_7\},4)$, $\sigma'_{T,5}(\{t_7\},5.7)$.

In Table II, for each step q of σ_T and of σ'_T, it is reported the set T_q of fired transitions, the set EM_{jq} of possible markings that enable the firing of each transition $t_j \in T_q$ (for which constraints (8) are written), the set of marking that surely do not enable the firing of each transition $t_j \in T_q$ (for which constraints (9) are written), called DM_{jq}.

TABLE II

FIRING TRANSITIONS, SET EM_{jq} OF POSSIBLE ENABLING MARKINGS AND SET DM_{jq} OF DISABLING MARKING FOR EACH TRANSITION OF THE SET OF THE OBSERVED TIMED FIRING SEQUENCE BELONGING TO $\mathcal{L}_{obs}$.

	q	T_q	EM_{jq}	DM_{jq}
σ_T	1	$\{t_7\}$	EM_{71}=$\{\boldsymbol{m}_0\}$	
	2	$\{t_1,t_2\}$	EM_{j2}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$, $j=[1,2]$	
	3	$\{t_3\}$	EM_{33}= $\{\boldsymbol{m}_0,\boldsymbol{m}_1,\boldsymbol{m}_2\}$	
	4	$\{t_4\}$	EM_{44}=$\{\boldsymbol{m}_2,\boldsymbol{m}_3\}$	DM_{44}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$
	5	$\{t_5,t_6\}$	EM_{j5}=$\{\boldsymbol{m}_3,\boldsymbol{m}_4\}$ $j=[5,6]$	DM_{j5}= $\{\boldsymbol{m}_0,\boldsymbol{m}_1,\boldsymbol{m}_2\}$, $j=[5,6]$
σ'_T	1	$\{t_7\}$	EM_{71}=$\{\boldsymbol{m}_0\}$	
	2	$\{t_2\}$	EM_{22}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$	
	3	$\{t_1\}$	EM_{13}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$	
	4	$\{t_3\}$	EM_{34}=$\{\boldsymbol{m}_2,\boldsymbol{m}_3\}$	DM_{34}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$
	5	$\{t_4\}$	EM_{45}= $\{\boldsymbol{m}_2,\boldsymbol{m}_3,\boldsymbol{m}_4\}$	DM_{45}=$\{\boldsymbol{m}_0,\boldsymbol{m}_1\}$
	6	$\{t_5,t_6\}$	EM_{j6}=$\{\boldsymbol{m}_4,\boldsymbol{m}_5\}$ $j=[5,6]$	DM_{j6}= $\{\boldsymbol{m}_0,\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}$ $j=[5,6]$

Notice that $L(\mathcal{T}) \supseteq \mathcal{L}_{obs}$ but the net has a smaller number of places and arcs than the one in Fig. 8(b).

For these nets it is possible to find at least one timed sequence that belongs to $\mathcal{L}_{unid}$, as example, the sequence $\hat{\sigma}_T = (\{t_7,0\})(\{t_1,t_2\},1.3) \in \mathcal{L}$ does not belong to $L(\mathcal{T})$, i.e., $\hat{\sigma}_T \in \mathcal{L}_{unid}$. On the other hand the identified system produces all sequences that belong to the unknown system too, i.e., $\mathcal{L}_{exc} = \emptyset$.

The *inferred* language $\mathcal{L}_{inf}$ is not empty, indeed the timed sequence $\tilde{\sigma}_T = (\{t_7,0\})$ $(\{t_1,t_2\},1)$ $(\{t_3\},3)$ $(\{t_4\},4)$ $(\{t_5,t_6\},5.8) \in L(\mathcal{T})$ belongs to $\mathcal{L}_{inf}$ too.

The size of $\mathcal{L}_{unid}$ and $\mathcal{L}_{inf}$ depends on the set of the observed sequences. As example if the new sequence $\sigma''_T = (\{t_7,0\})(\{t_1,t_2\},1)$ $(\{t_3\},3)$ $(\{t_4\},4)$ $(\{t_5,t_6,t_7\},6)$ $(\{t_2\},7)$ $(\{t_1\},7.2)$ $(\{t_3\},9.2)$ $(\{t_4\},10)$ $(\{t_5,t_6,t_7\},11.7)$ $(\{t_1,t_2\},13)$ $(\{t_3\},15)$ $(\{t_4\},16)$ $(\{t_5,t_6,t_7\},17.9)$ $(\{t_1\},19)$ $(\{t_2\},19.1)$ $(\{t_3\},21)$ $(\{t_4\},22.1)$ $(\{t_5,t_6\},24.1)$ is considered, the net of Fig. 8(d), is identified, for which $\mathcal{L}_{unid} = \emptyset$.

Notice that, even if a finite timed firing sequence has been used, a cyclical net has been obtained. This occurs because of the particular cost function used in the MILPP that tries to minimize the number of arcs.

VI. COMPUTATIONAL COMPLEXITY

The approach presented in this paper is based on the solution of a MILPP, whose complexity is known to be NP-hard. The focus of this section is the size of the MILPP (14).

Problem (14) can be characterized in terms of the number of constraints and unknowns that composed it, i.e., the number of constraints and unknowns of each $\mathcal{G}(\sigma_T)$ and $B(p_i)$.

Number of constraints and unknowns of the MILPP – system $\mathcal{G}(\sigma_T)$ and constraints $B(p_i)$ – depends on the following parameters:

- L = length of the timed firing sequence σ_T (i.e., the number of couples (T_q,τ_q) sequence σ_T is made up of).

- $n =$ cardinality of T.
- $m =$ number of places.
- $n_c =$ cardinality of T^c.

The number of constraints of $\mathcal{G}(\sigma_T)$ is given by the number of constraints of $\mathcal{C}(T_q, \tau_q)$ plus $n{+}2{\cdot}n_c{+}m{\cdot}(L{+}1)$ constraints due to equations (13b-13e). The number of constraints of $\mathcal{C}(T_q, \tau_q)$ is given by the sum of the number of constraints of systems $\mathcal{M}_e$, $\mathcal{M}_c$ and $\mathcal{M}_d$, named as $y_{\mathcal{M}_e}$, $y_{\mathcal{M}_d}$ and $y_{\mathcal{M}_c}$, respectively.

The set of unknowns of $\mathcal{G}(\sigma_T)$ is composed of two components: a) the set of N_a unknowns and b) the set of $u^{bv}_{\mathcal{G}(\sigma_T)}$ boolean variables.

Consequently the total number of unknowns of $\mathcal{G}(\sigma_T)$ is

$$u_{\mathcal{G}(\sigma_T)} = N_a + u^{bv}_{\mathcal{G}(\sigma_T)} \tag{22}$$

The set of unknowns consists of m integer unknowns, representing the initial marking $\boldsymbol{m}_0$ of the net, $2 \cdot n \cdot m$ integer unknowns, representing the $\mathbf{Pre}$ and $\mathbf{Post}$ incidence matrices of the net and $2 \cdot n$ real unknowns, representing the bounds l_j and u_j of the firing interval $I(t_j)$ of each transitions of the net.

As consequence

$$N_a = m + 2 \cdot (n \cdot m) + 2 \cdot n \tag{23}$$

The number of constraints introduced by $B(p_i)$ is:

$$y_{B(p_i)} = \quad (m \cdot n) + 2m \tag{24}$$

while the corresponding number of boolean variables is:

$$u^{bv}_{B(p_i)} = \quad (m \cdot n) \tag{25}$$

Therefore the total number of constraints of the identification problem (14) is given by

$$y_{\mathcal{M}_e} + y_{\mathcal{M}_d} + y_{\mathcal{M}_c} + n + 2 \cdot n_c + m \cdot (L + 1) + y_{B(p_i)}$$

while the total number of unknowns is given by

$$u_{\mathcal{G}(\sigma_T)} + u^{bv}_{B(p_i)}$$

In the following, the size of the problem associated to the example of Section V-G, when $\mathcal{L}_{obs} = \{\sigma_T, \sigma'_T\}$, is shown.

Values of parameters of the example are:

$$L_1 = 5,\ L_2 = 6,\ n = 7,\ n_c = 1,\ m = 7.$$

Each observed sequence belonging to $\mathcal{L}_{obs}$ introduces a certain number of constraints and boolean variables. Such a number is shown in Table III. The total number of constraints and variables is obtained by the sum of those introduced by each observed sequence.

By using a machine having Intel© Core™ i7 CPU at 2.67 GHz, 8.00 GB of Ram and a 64 bit operative system, resolution of (14) with Cplex©, running in Matlab© environment, takes 8.74 seconds. The same problem takes 17.43 seconds with $m = 9$.

The complexity of the proposed approach is not trivial and the most complex step is the solution of a MILPP. However, it is a standard tool, and so a lot of efficient solvers can be used to solve it, and so this makes it effective in practice, especially for workflow and manufacturing systems which are characterized by event interarrival times slower with respect to computer and communication systems. However, as written in Section V, the proposed approach can be used also to process a batch of observations. Moreover, this aspect is less relevant for its application to fault diagnosis/model repair where a fault-free/nominal system is available and only the faulty/new subsystem must be identified, so the MILPP is significantly less complex.

TABLE III
NUMBER OF CONSTRAINTS AND UNKNOWNS OF THE EXAMPLE, WITH $n = 7$, $n_c = 1$, $m = 7$.

	constraints		N_a	boolean variables	
	$L_1 = 5$	$L_2 = 6$		$L_1 = 5$	$L_2 = 6$
$\mathcal{M}_e$	1377	1476		350	350
$\mathcal{M}_d$	108	72		156	104
$\mathcal{M}_c$	1032	672		602	392
Eq.(13b)	7	7		0	0
Eq.(13c)	1	1		0	0
Eq.(13d)	1	1		0	0
Eq.(13e)	84	84		0	0
$B(p_i)$	63	63		49	49
			119		
total	2673	2376		1157	895

VII. CONCLUDING REMARKS

A mixed-integer linear programming approach for the identification of free labeled time PN models has been proposed. The presence of a timing structure requires not only that the firing of a transition is enabled but also that its firing interval is congruent with the time instant when the firing is observed. This has been taken into account by means of a set of logical propositions transformed into linear mixed-integer inequalities.

Future research efforts will focus on the extension of the results to labeled net systems and on their application to fault diagnosis and model repair of time Petri net systems.

REFERENCES

[1] A. P. Estrada-Vargas, E. Lopez-Mellado, and J.-J. Lesage, “A Comparative Analysis of Recent Identification Approaches for Discrete-Event Systems,” *Mathematical Problems in Engineering*, 2010.

[2] M. P. Cabasino, P. Darondeau, M. P. Fanti, and C. Seatzu, “Model identification and synthesis of discrete-event systems,” in *Contemporary Issues in Systems Science and Engineering*, ser. IEEE/Wiley Press Book Series, M. Zhou, H.-X. Li, and M. Weijnen, Eds. John Wiley & Sons, Inc., 2015, pp. 343–366.

[3] M. P. Cabasino, A. Giua, and C. Seatzu, “Identification of Petri nets from knowledge of their language,” *Discrete Event Dynamic Systems*, vol. 17, pp. 447–474, December 2007.

[4] K. Hiraishi, “Construction of a class of safe Petri nets by presenting firing sequences,” in *13th International Conference on Application and Theory of Petri Nets (ICATPN’92), Sheffield, UK*, ser. Lecture Notes in Computer Science, vol. 616. Springer, June 1992, pp. 244–262.

[5] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev, “Deriving Petri nets from finite transition systems,” *IEEE Trans. on Computers*, vol. 47, no. 8, pp. 859–852, August 1998.

[6] P. Darondeau, “Region Based Synthesis of P/T-Nets and Its Potential Applications,” in *21st International Conference on Application and Theory of Petri Nets 2000 (ICATPN 2000) Aarhus, Denmark*, ser. Lecture Notes in Computer Science, vol. 1825. Springer, June 2000, pp. 16–23.

[7] M. Meda-Campana and E. Lopez-Mellado, “Required event sequences for identification of Discrete Event Systems,” in *42nd IEEE Conference on Decision and Control, 2003*, vol. 4, Maui, Hawaii, December 2003, pp. 3778–3783.

[8] M. Dotoli, M. P. Fanti, and A. M. Mangini, “Real time identification of discrete event systems using Petri nets,” *Automatica*, vol. 44, no. 5, pp. 1209 – 1219, 2008.

[9] F. Basile, P. Chiacchio, J. Coppola, and G. De Tommasi, “Identification of Petri nets using timing information,” *3^{rd} International Workshop on Dependable Control of Discrete Systems (DCDS’11), Saarbrucken, Germany*, pp. 154 –161, 2011.

[10] A. Estrada-Vargas, E. Lopez-Mellado, and J.-J. Lesage, “Off-line identification of concurrent Discrete Event Systems exhibiting cyclic behaviour,” in *IEEE International Conference on Systems, Man and Cybernetics (SMC’09).*, October 2009, pp. 181 –186.

[11] IEEE Task Force on Process Mining, “Process mining manifesto,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Springer Berlin Heidelberg, August 2012, vol. 99, pp. 169–194.

[12] W. M. van der Aalst, “Process mining in the large: A tutorial,” in *Business Intelligence: 3rd European Summer School, eBISS 2013, Dagstuhl Castle, Germany*, ser. Lecture Notes in Business Information Processing, E. Zimányi, Ed. Springer International Publishing, July 2014, vol. 172, pp. 33–76.

[13] W. van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: discovering process models from event logs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, September 2004.

[14] F. Basile, P. Chiacchio, and J. Coppola, “A hybrid model of complex automated warehouse systems - part II: Analysis and experimental results,” *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 654–668, October 2012.

[15] F. Basile, P. Chiacchio, and D. Teta, “A hybrid model for real time simulation of urban traffic,” *Control Engineering Practice*, vol. 20, no. 2, pp. 123–137, February 2012.

[16] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. Springer–Verlag London, 2009.

[17] C. Ramchandani, “Analysis of asynchronous concurrent systems by timed Petri nets,” Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1974.

[18] P. M. Merlin, “A study of the recoverability of computing systems.” Ph.D. dissertation, University of California, Irvine, 1974.

[19] W. Khansa, J.-P. Denat, and S. C. Dutilleul, “P-time petri nets for manufacturing systems,” *Proceedings of the International Workshop on Discrete Event Systems (Wodes’96)*, 1996.

[20] H. Hanisch, “Analysis of place/transition nets with timed-arcs and its application to batch process control,” in *International Conference on Application and Theory of Petri Nets (ICATPN’93)*, ser. Lecture Notes in Computer Science, M. A. Marsan, Ed. Springer Berlin Heidelberg, 1993, vol. 691, pp. 282–299.

[21] M. Boyer and O. Roux, “Comparison of the expressiveness of arc, place and transition time petri nets,” in *Petri Nets and Other Models of Concurrency, ICATPN 2007*, ser. Lecture Notes in Computer Science, J. Kleijn and A. Yakovlev, Eds. Springer Berlin Heidelberg, 2007, vol. 4546, pp. 63–82.

[22] S. Ould El Mehdi, R. Bekrar, N. Messai, E. Leclercq, D. Lefebvre, and B. Riera, “Design and identification of stochastic and deterministic stochastic Petri nets,” *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 4, pp. 931–946, 2012.

[23] M. Meda-Campana and S. Medina-Vazquez, “Synthesis of timed Petri net models for on-line identification of discrete event systems,” *9th IEEE International Conference on Control and Automation (ICCA’11), Santiago, Chile*, pp. 1201–1206, 2011.

[24] J. Peterson, *Petri Net Theory and the Modeling of Systems*, P. Hall, Ed., 1981.

[25] B. Hrúz and M. Zhou, *Modeling and Control of Discrete Event Dynamic Systems: With Petri nets and other tools*. Springer–Verlag London, UK, 2007.

[26] F. Basile, P. Chiacchio, and J. Coppola, “An approach for the identification of time Petri net systems,” in *2013 IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), Cagliari*, September 2013, pp. 1–8.

[27] J. Zaytoon and S. Lafortune, “Overview of fault diagnosis methods for Discrete Event Systems,” *Annual Reviews in Control*, vol. 37, pp. 308 – 320, 2013.

[28] H. Hu, Z. Li, and A. Al-Ahmari, “Reversed fuzzy petri nets and their application for fault diagnosis,” *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 505 – 510, 2011.

[29] M. Cabasino, A. Giua, C. N. Hadjicostis, and C. Seatzu, “Fault model identification and synthesis in Petri nets,” *Discrete Event Dynamic Systems*, pp. 1–22, 2014.

[30] G. Chang and D. Kulic, “Robot task error recovery using Petri nets learned from demonstration,” in *16th International Conference on Advanced Robotics (ICAR), 2013*, November 2013, pp. 1–6.

[31] A. Rogge-Solti and G. Kasneci, “Temporal anomaly detection in business processes,” in *Business Process Management: 12th International Conference, BPM 2014, Haifa, Israel*, ser. Lecture Notes in Computer Science, S. Sadiq, P. Soffer, and H. Völzer, Eds. Springer International Publishing, September 2014, vol. 8659, pp. 234–249.

[32] T. Murata, “Petri nets: Properties, analysis and applications,” *Proceedings of IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.

[33] C. Seatzu, M. Silva, and J. H. van Schuppen, Eds., *Control of Discrete-Event Systems: Automata and Petri Net Perspectives*, ser. Lecture Notes in Control and Information Sciences. Springer–Verlag London, 2013, vol. 433.

[34] S. Klein, L. Litz, and J.-J. Lesage, “Fault detection of discrete event systems using an identification approach,” in *16th IFAC world Congress, Praha, Tchèque, République*, 2005, p. CDROM paper n. 02643.

[35] M. Ajmone Marsan, A. Bobbio, and S. Donatelli, “Petri nets in performance analysis: An introduction,” in *Lectures on Petri Nets I: Basic Models*, ser. Lecture Notes in Computer Science, W. Reisig and G. Rozenberg, Eds. Springer Berlin Heidelberg, 1998, vol. 1491, pp. 211–256.

[36] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, pp. 407–427, 1999.