A novel model repair approach of timed discrete event systems with anomalies

Francesco Basile, Pasquale Chiacchio, Jolanda Coppola

Abstract— In this paper the model repair of timed discrete event systems where anomalies may occur is considered. The nominal model is assumed to be known and a set of observed timed sequences is given. The approach works with Time Petri net models and is based on the formulation of a Mixed-Integer Linear Programming Problem. The repaired model is obtained from the nominal one by adding fault transitions as well as by extending the firing interval of transitions.

Note to Practitioners: The data collected from the observation of a discrete event system are usually given in terms of behavioral sequences that may be fixed or may be increased in the course of the system operation (e.g., due to new experiments or simply to the system running). If this occurs, the nominal model of a system may reveal not consistent with these additional observations if they include anomalies. This work presents an approach to repair the model of the system in order to make it able to also generate the observed timed faulty behavior. The repaired model can be used to perform fault diagnosis.

Keywords: Petri Nets, Discrete Event Systems, Identification, Model repair.

I. INTRODUCTION

Automated modeling 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. This problem has been addressed by the Discrete Event Systems (DESs) and Workflow Management Systems communities, under different names (DES Identification and Process mining, respectively) and approaches. Although there are several proposed approaches in each community, much remains to be done regarding the modeling of timed discrete event processes/systems.

The explicit consideration of time is crucial for the specification and the verification of some DESs [1] such as communication protocols, circuits, urban traffic [2] or realtime systems and automated manufacturing systems. Two main techniques were developed from Petri Nets (PNs): timed PNs [3] and time PNs [4]. In the first, a fixed firing duration is associated with each transition while, in the second, the firing duration of a transition t can assume any value of a given interval I(t).

Postprint - Work published on IEEE Transactions on Automation Science and Engineering (http://dx.doi.org/10.1109/TAC.2014.2363916)

Process mining aims to discover, monitor and improve real processes by extracting knowledge from a collection of sequential events and information about the system [5], [6].

A particular kind of process mining is the *Model repair*: it consists in modifying the nominal model of a system as a consequence of the occurrence of the observation of discrepancies between the system nominal behavior and the system observed behavior, in the manner that the modified model completely describes the observed behavior. These discrepancies, named *anomalies*, can be due to different reasons: workers start handling activity differently, system components degrade, action of external agents, etc. The occurrence of one of these circumstances modifies the system dynamic as well as the duration of the activities of the system and, as a consequence, the nominal model needs to be modified.

Model repair has been introduced for the first time in [7]. In such a work, the new model (i.e., the "repaired" one) is obtained adding new subprocesses to the nominal model in the manner that the resulting model fits the observed behavior and it is as similar as possible to the original one.

The problem of modifying the nominal model as a consequence of changes in the system behavior has been investigated in the field of DES identification too, and in particular it has been treated in [8]. In such a work, anomalies are called *faults* and the model repair is presented as the identification of the *faulty model* of a logical PN system: the occurrence of a faulty firing sequence (i.e., a sequence that cannot be generated by the nominal model of the system) is associated to the unobservable firings of fault transitions, that must be opportunely added and linked to the nominal model of the system, to obtain the faulty model. Hence, also in this case, the structure of the nominal model is changed.

The most trivial solution to this problem, discussed in [8], is obtained by adding one fault transition with no input place and connected to all places that are input places for the transitions of the nominal model. This is not significant in real applications since it only enables additional sequences that were forbidden with no fault, but it does not simultaneously disable other sequences, and this is a crucial effect of a fault occurrence.

Contribution of this paper is to extend the model repair approach to timed DESs, obtaining a repaired model able to generate the observed timed faulty behavior of the system.

Based on the time information, the faulty behavior of a system leads to i) an unusual activity duration, that can be due to a deterioration of performances or to a non-optimal execution of operations; ii) a change of the system dynamic that can be due to hardware failures as well as to a wrong execution of operations. In both cases the effect of the occurrence of a fault is that, at a given time, some unexpected events

Francesco Basile, Pasquale Chiacchio and Jolanda Coppola are with Dip. Ingegneria dell'Informazione, Ingegneria elettrica e Matematica applicata, Univ. di Salerno, Italy.

^{©2014} 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.

occur, while some expected others do not. This is the key concept used in the paper to formulate a Mixed-Integer Linear Programming Problem (MILPP), whose solution provides the corrections needed to repair the nominal model, and precisely the extension of the firing interval of the nominal transitions, i.e., an extension of the bounds of the firing time of each nominal transition, and/or the addition of fault transitions.

Indeed, these corrections identify the repaired model according to the observed behavior, so the proposed approach can be considered as a particular case of net identification.

A. Relevant literature

The interest for the identification of DESs usually comes from reverse engineering for (partially) unknown systems, fault diagnosis, or system verification. Inputs and/or outputs 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 PN or a finite state automaton model of the system behavior from sequences observed during the system operation [9], [10]. 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 [11], [12], [13], [14]. 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 [15], [16], [17]. In such a framework the main goal is to periodically execute an identification algorithm that provides a model able to generate the observed strings.

In some cases the net identification is accomplished not only on the basis of the *event observation*, [15], [17], [18], but also observing the net marking (*marking observation*) [16], [19], [20].

In [20] the identification of the unobservable behavior of PN models is considered. Apart from the fact that untimed models are considered, the main difference with respect to our approach is that it is based on event observation and marking observation, as well.

While a rich literature exists on the problem of the identification of *logical PNs* (for example [16], [15], [21], [8], [22], [23]), to the best of authors' knowledge, only few works have been published on the identification of *timed net systems* [18], [24], [19], [17]. The problem discussed in this paper can be considered as a net identification problem based only on event observation.

In [25] authors present a method to identify a time PN modeling the system on the basis of the observed behavior. The problem addressed here is different from [25], since a model repair problem is addressed and the identification regards only the subnet modeling the timed faulty behavior of the system

while the subnet modeling the timed nominal behavior is assumed to be known.

II. NOTIONS AND ASSUMPTIONS

A. Background on Petri nets

For a complete review on PNs the reader can refer to [26]. 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), $\mathbf{Pre} : P \times T \to \mathbb{N}$ (**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*); $C = \mathbf{Post} - \mathbf{Pre}$ is the incidence matrix.

A marking is a function $m : P \to \mathbb{N}$ 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 $m \in \mathbb{N}^m$. A net system $S = \langle N, m_0 \rangle$ is a net N with an initial marking m_0 . A transition t is enabled at miff $m \ge \operatorname{Pre}(\cdot, t)$ and this is denoted by $m[t\rangle$. An enabled transition t may fire yielding the marking $m' = m + C(\cdot, t)$ and this is denoted by $m[t\rangle m'$.

A firing sequence from m is a sequence of transitions $\sigma = t_1 \dots t_k$ such that $m[t_1\rangle m_1[t_2\rangle m_2 \dots [t_k\rangle m_k$, and this is denoted by $m[\sigma\rangle m_k$. An enabled sequence σ is denoted by $m[\sigma\rangle$, while $t_j \in \sigma$ denotes that transition t_j belongs to the sequence σ . A marking m' is said to be reachable from m_0 iff there exists a sequence σ such that $m_0[\sigma\rangle m'$. $R(N, m_0)$ denotes the set of reachable markings of the net system $\langle N, m_0 \rangle$.

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

The function $\sigma: T \to \mathbb{N}$, where $\sigma(t)$ 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 $\sigma \in \mathbb{N}^n$. Note that, if a sequence is made up of a single transition, i.e., $\sigma = t_j$, then the corresponding firing count vector is the *j*-th canonical basis vector denoted as e_j .

If $m_0[\sigma\rangle m$, then it is possible to write in vector form

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

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

Definition 1 (Time Petri net system, [27]): Let \mathcal{I} be the set of closed intervals with a lower bound in the set of positive rational numbers \mathcal{Q}^+ and an upper bound in $\mathcal{Q}^+ \bigcup \infty$. A Time Petri net (TPN) system is the triple $S = \langle N, \boldsymbol{m}_0, I \rangle$, 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 $[l_i, u_i]$ to each transition $t_i \in T$.

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, an enabled transition must fire if the upper bound of $I(t_j)$ is reached, thus enforcing urgency. A clock measuring the time elapsed from the enabling is implicitly associated to any transition.

It is assumed that there is a start-up transition that fires only once at time zero producing tokens considered by the initial marking and setting to zero the value of clocks. Given a set S, |S| denotes the cardinality of S.

B. Assumptions

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

 \Diamond

- 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 to each transition t is assigned a label, and the firing of t is an event that generates the label as observable output. This assumption restricts to free labeled nets the net subclass that the proposed approach can identify, 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) *Single-server* firing semantic (more details in [27]), i.e., no concurrent firings of the same transition are possible.
- 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.

For a better presentation of the approach proposed in this paper the definition of timed firing sequence \mathfrak{S} is needed. With this aim, it is useful to collect in the same set all those transitions firing at the same time τ_a .

Definition 2 (Timed firing sequence): A sequence

$$\mathfrak{S} = (T_1, \tau_1) \dots (T_q, \tau_q) \dots (T_L, \tau_L),$$

where T_q is the set of transitions fired at time τ_q and $\tau_1 < \tau_2 \cdots < \tau_L$ denote firing time instants, is called *timed firing sequence*. The position q that 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 \mathfrak{S} is called length $L = |\mathfrak{S}|$ of the timed firing sequence.

The notation $m[\mathfrak{S}\rangle m'$ is used to denote that m' is reached from m by firing \mathfrak{S} .

Definition 3 (Timed Language): Given a TPN system $S = \langle N, m_0, I \rangle$, its timed language, named $\mathcal{L}(S)$, is defined as the set of timed firing sequences generated by S from the initial marking m_0 .

The marking the system reaches after the firing of all the transitions in T_q is called m_q .

This paper focuses on the context of automated manufacturing systems, where a control architecture interacts with a plant according to a scan time faster than the time evolution of the system. In this context, the multiple firing of a transition at the same time instant has no sense. This motivates the next assumption.

Assumption 2: A transition can fire only once in the same time instant.

However, the results presented in this paper are still valid removing this assumption, introducing some technicalities.

The set T_q is made up of $n_q = |T_q|$ transitions whose firing is observed at the same instant τ_q . The firings of these transitions are enabled either by a marking m_k reached at a



Fig. 1. (a)-(c) Evolution of the net system; (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 instants for immediate firings of transitions; (e) firing sequence for T_3 .

time $\tau_k < \tau_q$ or by the firing of another transition fired at τ_q with null firing duration.

Definition 4 (Firing Duration): Given a timed transition t_j , fired at the q-th step, enabled at the k-th step, so that $m_k[t_j\rangle$, let m_k be the first marking that enables t_j since its previous firing, the function $\delta(t_j, k, q) : T \times \mathbb{N} \times \mathbb{N} \to Q^+$ 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 to as the firing duration of transition $t_j \in T_q$ from the marking m_k . When $\delta(t_j, k, q) =$ 0 the firing of t_j at τ_q is called *immediate*, otherwise, when $\delta(t_j, k, q) > 0$, the firing of t_j is called *timed*.

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 $\mathcal{M}(t_j,q) = \{\boldsymbol{m}_k \mid \exists \mathfrak{S}'_T, \mathfrak{S}''_T, \mathfrak{S} = \mathfrak{S}'_T \mathfrak{S}''_T, \boldsymbol{m}_0[\mathfrak{S}'_T \rangle \boldsymbol{m}_k[\mathfrak{S}''_T \rangle \boldsymbol{m}_q, \text{ with } t_j \in \mathfrak{S}''_T, k < q : \tau_k + l_j \leq \tau_q \leq \tau_k + u_j \}$, having cardinality $|\mathcal{M}(t_j,q)|$.

 $q: \tau_k + l_j \leq \tau_q \leq \tau_k + u_j$, having cardinality $|\mathcal{M}(t_j, q)|$. The set T_q can be partitioned into the couple of sets $(T_q^t, T_q^{im}): T_q^t = \{t_j \in T_q | \exists k, m_k \in \mathcal{M}(t_j, q)\}$ 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 transitions fired at τ_q with immediate firing.

Immediate firings always follow the timed ones, even if they are observed at the same time τ_q . Indeed an immediate firing occurs at the same time it has been enabled, while a timed firing occurs in a subsequent time respect the one at which it has been enabled. Consequently, a timed firing enabled by an immediate firing occurred at time τ_q , surely fires in a time greater than τ_q .

The firing of transitions in the set T_q^t is concurrent, however, each firing can have been enabled at a different marking. As an example, consider the system of Fig. 1(a) and assume that the sequence $\mathfrak{S}=(\{t_1\}, \tau_1)$ $(\{t_2\}, \tau_2)$ $(\{t_3, t_4, t_5, t_6\}, \tau_3)$ is observed. At step 3, on the basis of the nominal model of the system, the set T_3 can be decomposed in the couple of sets $(T_3^t = \{t_3, t_5\}, T_3^{im} = \{t_4, t_6\}).$

As shown in Fig. 1(d), the firings of t_5 and t_3 have been enabled at m_1 (Fig. 1(b)) and m_2 (Fig. 1(c)), respectively, but, since the firing duration of t_5 from m_1 is equal to 2 while $\delta(t_3, 2, 3) = 1$, their firings are observed at the same time τ_3 .

Denote m_{q_1} the marking reached by firing transitions belonging to T_q^t , i.e., with reference to the example of Fig. 1, after the firing of transitions $\{t_3, t_5\} \in T_3^t$ marking m_{3_1} is reached.

On the other hand, the firing of transitions in T_q^{im} may be sequential. Given the set of transitions T_q^{im} , these transitions can fire in any order, which, anyway, can include concurrent transition firings. Denote m_{q_s} , with $s \ge 2$, the marking reached after the immediate firings of transitions.

Given the firing sequence associated to the set T_q^{im} , it can be considered made up of the union of n_q^{im} disjoint subsets of concurrent transition firings. Hence, firing of transitions in T_q^{im} can be considered occurred in n_q^{im} substeps; each substep is denoted by q_s , with $s \in [2, n_q^{im} + 1]$. Finally, it holds that $T_q^{im} = \bigcup_{s=2}^{n_q^{im}+1} T_{q_s}^{im}$. As indicated in Fig. 1(e), the marking reached after the

As indicated in Fig. 1(e), the marking reached after the firing of transitions in the last of these subsets corresponds to m_q ; moreover, the firing sequence associated to $T_3 = \{t_3, t_4, t_5, t_6\}$ is shown. In detail, $T_3^t = \{t_3, t_5\}$, and $T_2^{im} = \{t_4, t_6\}$; moreover, $T_{32}^{im} = \{t_4\}$, $T_{33}^{im} = \{t_6\}$. In figure, triangles indicate the firing of timed transitions that leads to the reaching of the first marking m_{31} , while diamonds indicate the immediate firings of transitions in T_3^{im} .

III. PROBLEM FORMULATION

Goal of this work is to repair the system model identifying the subnet modeling the observed faulty behavior. Such a subnet is added to the system nominal model in order to obtain a system repaired model, that is assumed as the current model of the system. With this aim, two techniques are used:

- a) the extension of the transition firing intervals;
- b) the addition of fault transitions.

As far as the observed behavior of the system where anomalies occur, at a given time, *unexpected firings* of transitions as well as *missing firings* of transitions, incoherent with respect to the current model, are observed. The firing of a transition t_j at time τ_q is called unexpected if 1) it fires after a time less than its lower bound from its enabling, 2) it fires after a time greater than its upper bound or 3) there does not exist a reachable marking in the current model under which its firing at τ_q could have been enabled. A missing firing of t_j at τ_q occurs when t_j does not fires at τ_q even though it has been enabled for a time equal to u_j .

The technique a) can be applied when an unexpected firing of type 1) or 2) occurs. In this case, the firing of a transition t_j , enabled at a marking m_{k_s} reached at a time $\tau_k \leq \tau_q$, occurs at a time τ_q such that $\tau_q - \tau_k \notin I(t_j)$. Unexpected firings that can be modeled as firing interval extensions are also called *temporal anomalies*. A large set of failures can be modeled as



Fig. 2. (a) Nominal TPN; (b) faulty TPN.

temporal anomalies (a slowing down of a conveyor belt speed due to the wear, a shorter duration of a work phase due to an incorrect handling of the operator, a casual change in a time duration, etc.).

However, in practice there are some other failures, for example, breakdowns or changes in work phase sequence, that cannot be modeled just extending the transition firing interval. To model this kind of anomalies, the net structure must be changed, for example adding fault transitions, and so the technique b) can be used. Fault transitions are unobservable, since their firing cannot be observed, and can be used to generate the faulty behavior. Their firings modify the marking of the system in the way that the enabling of some transitions is anticipated, leading to their unexpected firings of type 1), or some disabled transitions becomes enabled, leading to their unexpected firing of type 3), or some enabled transitions become disabled, leading to their missing firings. Then, introduction of fault transitions is much more powerful than extending the firing interval of transitions.

The proposed approach is able to detect anomalies and to repair the nominal model faster than existing logical approaches, as shown in the following simple motivational example.

Consider the TPN in Fig. 2(a): the timed firing sequence $\mathfrak{S}=(\{t_1\}, 1)$ $(\{t_1\}, 3)$ $(\{t_2\}, 5)$ $(\{t_2\}, 7)$ is a faulty sequence since at $\tau_4 = 7$ the missing firing of transition t_3 (that has been enabled by the firing of transition t_2 at time $\tau_3 = 5$) and the unexpected firing of t_2 occur.

Notice that the firing of transition t_2 at time $\tau_4 = 7$ is an unexpected firing because a time less than $l_2 = 3$ is elapsed from its previous firing.

The proposed approach, on the basis of the detected anomalies, returns the system in Fig. 2(b) as a possible faulty model where a fault transition has been added and the firing interval of t_2 has been enlarged; existing approaches only consider the logical behavior of the system, so they cannot detect the fault occurrences at the step 4. Indeed the logical firing sequence $\sigma = t_1 t_1 t_2 t_2$, obtained from \mathfrak{S} discarding the time information, belongs to the behavior of the nominal model. Only if a third firing of t_2 is observed, it will be possible to conclude that a fault is occurred.

For a better presentation of the repaired model identification algorithm in the next section, it is convenient to formally characterize the faulty behavior of system at time τ_q .

A system presents a faulty behavior at time τ_q if some unexpected as well as some missing firings occur at τ_q .

At each time τ_q concurrent firings of different transitions can occur. Some of these firings are justified by the current model of the system while the remaining part is caused by the occurrence of faults. Thus, given the couple (T_q, τ_q) , the set



Fig. 3. (a) System of Example 1: two cars going towards right and returning, passing through a crossroad; (b) TPN modeling the nominal behavior.

TABLE I TIME NEEDED TO THE CARS OF EXAMPLE 1 TO RUN ALONG EACH PATH.

| Path | t.u. |
|---------------------------|------|
| $a - h_{1A} (b - h_{2A})$ | 1 |
| $a - h_{1B} (b - h_{2B})$ | 1.3 |
| $h_{1B} - f (h_{2B} - e)$ | 1 |
| e - d | 1.5 |
| f - c | 0.5 |
| d - a | 3.5 |
| c-b | 0.5 |

 T_q can be partitioned in the set T_q^{un} - the set of transitions for which an unexpected firing occurred at time τ_q , and in the T_q^n - the set of transitions for which the firing at time τ_q is coherent with the current model.

The set T_q^{miss} collects all those transitions for which a missing firing occurred at time τ_q .

Consequently, the faulty behavior of the system at time τ_q is characterized by the triple $(T_q^{un}, T_q^{miss}, \tau_q)$. Notice that $T_q^{un} \cap T_q^{miss} = \emptyset$.

In general, a missing firing of a transition and an unexpected firing of another transition may be simultaneous. When only a missing firing is observed, the term time-out is also used to denote that nothing else is observed but a transition, enabled for a time equal to the upper bound of its firing interval, does not fire. In this situation, the only reasonable repair action is the adding of a fault transition. Indeed, to enlarge the firing interval of a transition its effective occurrence instant must be known, but this is unknown at the time the time-out is detected.

Hereinafter, given a timed firing sequence S $(T_1, \tau_1) \dots (T_q, \tau_q) \dots (T_L, \tau_L)$, it is assumed that the set T_q can be empty too, i.e., $T_q = \emptyset$, to represent the case where it has not been observed the firing of any transitions at time τ_a , even thought some of them have been enabled at a time equal to their firing interval upper bound, so a time-out is occurred. Obviously, if $T_q = \emptyset$ and $m_{q-1}[(T_q, \tau_q) > m_q)$, then $m_q = m_{q-1}$. This is a technical extension of Definition 2, useful in the identification algorithm.

Example 1: The considered example is an adapted version of the system used in [18]: it is a system made up of two cars C1 and C2 (Fig. 3(a)), that starting from an arbitrary position in the home space (delimited by points h_{1A} and h_{1B} for C1

TABLE II MEANING OF TRANSITIONS OF EXAMPLE 1.

| Transition | Event |
|-------------|---|
| $t_1 (t_2)$ | C1 (C2) has arrived at a (b). |
| $t_3 (t_4)$ | C1 (C2) has entered in the crossroad. |
| $t_5 (t_6)$ | C1 (C2) has arrived at $f(e)$. |
| $t_7 (t_8)$ | C1 (C2) has arrived at c (d). |
| t_9 | Both cars have returned in their home position. |
| t_{10} | Both cars are ready to start again. |
| t_{11} | Cycle starts again. |

and h_{2A} and h_{2B} for C2, in the figure) 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 run along each path are reported in Table I. The cars must pass through the crossroad. To avoid collisions only one car at time can pass through it, consequently C1 or C2 must halt until the way is free. Once arrived at point c (d), C1 (C2) stops and remains in this state until both cars are in their right positions. It takes from 4.5 to 4.8 t.u. to return cars in home position, through the external path (after any travel, C1 and C2 trade places with each other), then a new cycle is immediately started.

The TPN modeling the nominal behavior of such a system is shown in Fig. 3(b); in Table II the meaning of each transition is reported.

Assume that first the sequence $\mathfrak{S} = (\{t_{11}\}, 0) \ (\{t_2, t_4\}, 1)$ $({t_1}, 1.2)$ ($\emptyset, 2$) and then, after the restart of the system from its initial condition, the sequence $\mathfrak{S}' = (\{t_{11}\}, 0) \ (\{t_1, t_3\}, 1)$ $(\{t_2, t_4\}, 1.3)$ have been observed.

Sequence \mathfrak{S} is a faulty sequence since the missing firing of t_6 is observed at time τ_4 , thus $(T_q^{un} = \emptyset, T_q^{miss} = \{t_6\}, \tau_q =$ au_4). In detail, the missing firing of t_6 is a time-out since no other faults have occurred at the same time.

Sequence \mathfrak{S}' is a faulty sequence since an unexpected firing of t_4 is observed at time τ_3 . Consequently, $(T_q^{un} =$ $\{t_4\}, T_q^{miss} = \emptyset, \tau_q = \tau_3$). The occurrence of \mathfrak{S} can have been caused by a slowdown of C2 which consequently needs more than 1 t.u. to run the crossroad. The occurrence of \mathfrak{S}' can be due to the transit of C2 along the crossroad before C1 has arrived at point e. \Diamond

IV. REPAIRED MODEL IDENTIFICATION ALGORITHM

The proposed algorithm to identify the repaired model of a system is shown in Fig. 4.

Let $S_0 = \langle N, \boldsymbol{m}_0, I \rangle$ be the nominal model of the observed system: when the algorithm starts, the nominal model is assumed as the current model, named S.

Any time a new couple (T_q, τ_q) is acquired, the current timed firing sequence \mathfrak{S} is extended, queueing (T_a, τ_a) to the previous observations \mathfrak{S}_{prev} (initially $\mathfrak{S}_{prev} = \varepsilon$, i.e., it is the empty sequence). Successively it is tested if Scan generate \mathfrak{S} : if not, the current model of the system is replaced with the repaired model of the system, named S, to be identified otherwise observation continues. The identification of the repaired model is based only on the nominal model and on the observations, because the identification approach is not incremental, so, the current model of the system is used



Fig. 4. Repaired Model Identification Algorithm.



Fig. 5. A possible repaired model of the system of Fig. 2(a), able to generate the sequence $\mathfrak{S} = (\{t_1\}, 1)$ $(\{t_1\}, 3)$ $(\{t_2\}, 5)$ $(\{t_2\}, 7)$ $(\{t_3\}, 7.5)$

only to detect anomalies while it is useless with regard of the identification procedure.

Then, on the basis of the reparation, the level of criticality of the occurred anomaly is evaluated (by a human operator, by automated software routines, etc.). Such an evaluation is outside the scope of this paper, for example to enlarge the firing interval of a transition beyond a certain threshold may be considered an high level of criticality. In the case of high criticality the system is stopped: if it can be reset, i.e., its initial condition can be physically restored, \mathfrak{S} is added to \mathcal{L}^F_{obs} , the faulty language of the system, and a new observation starts, otherwise the algorithm ends.

Language $\mathcal{L}_{obs}^{\vec{F}}$ is composed only by faulty observed sequences, i.e., sequences of the kind $\mathfrak{S} = \mathfrak{S}_{prev}(T_q, \tau_q)$ for which at time τ_q an anomalous behavior of the system is detected, thus $T_q^{un} \neq \emptyset$ or $T_q^{miss} \neq \emptyset$.

The repaired model of the system, \hat{S} , is identified on the basis of the nominal model S_0 , the current observation \mathfrak{S} and of the faulty language \mathcal{L}_{obs}^F by solving a MILPP obtained transforming the logical conditions, presented in the following sections, into algebraic linear constraints.

Example 2: Consider again the system of Fig. 2. As previously stated, the observation of the faulty sequence $\mathfrak{S} = (\{t_1\}, 1) \ (\{t_1\}, 3) \ (\{t_2\}, 5) \ (\{t_2\}, 7)$ leads to the identification of the repaired model shown in Fig. 2(b), obtained by adding the fault transition t_{f1} to the nominal model and extending the firing interval of transition t_2 .

Coherently with the repair model identification algorithm, after the repaired model of Fig. 2(b) is identified, it is assumed as the current model of the system and it is used to test the

occurrence of new anomalies.

Assume now that the new couple $(\{t_3\}, \tau_5)$ is acquired and so the observed sequence becomes $\mathfrak{S} = \mathfrak{S}_{prev}(\{t_3\}, \tau_5) =$ $(\{t_1\}, 1)$ $(\{t_1\}, 3)$ $(\{t_2\}, 5)$ $(\{t_2\}, 7)$ $(\{t_3\}, \tau_5)$. Notice that the model of Fig. 2(b) is able to generate the sequence $\mathfrak{S} = \mathfrak{S}_{prev}(\{t_3\}, \tau_5)$ if $\tau_5 \in [8, 9]$. Hence not the simply observation of the firing of t_3 but also the time instant when it occurs is crucial to classify such an occurrence as an anomaly and consequently to start the identification of a new repaired model.

Here, it is assumed that $\tau_5 = 7.5$ and so the algorithm considers the firing of t_3 at time $\tau_5 = 7.5$ as an unexpected event because the current model in Fig. 2(b) is not able to generate it. Consequently, a new repaired model \tilde{S} must be identified.

A possible repaired model is the one shown in Fig. 5, where the fault transition added in the model of Fig. 2(b) has disappeared while the firing interval of t_3 has been enlarged, in detail its upper bound has been increased by $\tau_5 - 7 = 0.5$.

A. Acquiring of a new couple (T_q, τ_q)

At each step q a new couple (T_q, τ_q) is obtained by means of the following algorithm.

Algorithm 1: Acquiring of a new couple (T_q, τ_q) .

Let $timer_j$ be a timer variable associated to transition $t_j \in T$, such that at step q, $timer_j = -1$ if t_j is disabled and $timer_j = \tau_q - \tau_k$ if t_j is enabled, where τ_k is the enabling instant of t_j .

- Step 1: If a new couple (T_q, τ_q) has been observed then jump to Step 3.
- Step 2: If there exists a transition $t_j \in T$ such that $timer_j \ge u_j$ then $T_q := \emptyset$ and $\tau_q := \tau_k + u_j$.

Step 3: Return
$$(T_q, \tau_q)$$
.

In words, at each step q, on the basis of the current model of the system, it is known which are the enabled transitions and which is their enabling instant. While it is waiting for the observation of a new couple (T_q, τ_q) , the algorithm checks that no time-outs occur, i.e., it controls that there does not exist a transition t_j for which a time equal to u_j is elapsed from its enabling without that transition has fired.

If this occurs and no couples have been observed, a special couple having $T_q = \emptyset$ and $\tau_q := \tau_k + u_j$ is created, as a consequence.

B. Testing of the occurrence of an anomaly

The testing of the occurrence of an anomaly at time τ_q is carried out by testing if an unexpected firing or a missing firing has occurred.

The firing of a transition t_i is an unexpected firing if:

a) there does not exist a marking m_{k_s} , reached at time $\tau_k \leq \tau_q$ such that the marking enables the firing of t_j , thus m_{k_s} does not satisfy the following equation:

$$m_{k_s} = m_0 + (\text{Post} - \text{Pre}) \cdot \mathfrak{S}_{k_s} \ge \text{Pre}(\cdot, \mathbf{t_i}), \quad (2)$$

where \mathfrak{S}_{k_s} is the timed firing sequence obtained terminating \mathfrak{S} at the substep k_s and $\mathfrak{S}_{\mathbf{k}_s}$: $\mathbf{T} \to \mathbb{N}$ is its firing count vector, with $\mathfrak{S}_{k_s}(t_j)$ the number of occurrence of t_j in \mathfrak{S}_{k_s} ;

b) there exists a marking m_{k_s} , reached at time $\tau_k \leq \tau_q$ such that eq. (2) holds but time τ_q is such that $\tau_q - \tau_k < l_j$ or $\tau_q - \tau_k > u_j$.

A missing firing of $t_j \notin T_q$ at τ_q occurs when t_j is enabled at a marking m_{k_s} reached at a time $\tau_k \leq \tau_q$ but t_j does not fire at time $\tau_q = \tau_k + u_j$. Consequently, a transition t_j belong to T_q^{miss} if equation (2) holds and moreover $\tau_q = \tau_k + u_j$.

On the basis of these considerations the occurrence of an anomaly can be tested by means of the two following algorithms.

Algorithm 2: Testing of an unexpected firing occurrence.

For each transition $t_j \in T_q$

- Step 1: Let τ_{prev_j} be the time of the last firing of t_j before τ_q : if such a time does not exist, i.e., t_j fires for the first time at τ_q , then $\tau_{prev_j} := 0$.
- Step 2: Collect in the set named $\mathcal{M}_{en}(t_j, q)$, each marking $m_{k'_s}$ reached at a time $\tau_{k'}$ such that $\tau_{k'} \geq \tau_{prev_j}$ and τ'_k : $\tau_q - \tau'_k \leq l_j$, for which condition (2) holds and $\nexists \tau_{k''_s}$: $\tau_{k'_s} < \tau_{k''_s} < \tau_q$ s.t. $m_{k''_s}$ does not enable t_j . The markings in the set $\mathcal{M}_{en}(t_j, q)$ are candidate markings for the enabling of $t_j \in T_q$.
- Step 3: If $\mathcal{M}_{en}(t_j, q) = \emptyset$ then jump to Step 9.
- Step 4: Choose m_{k_s} as the oldest marking of $\mathcal{M}_{en}(t_j, q)$, thus $m_{k_s} \in \mathcal{M}_{en}(t_j, q)$ and $k_s = \min_{\forall k'_s \text{ s.t. } m_{k'_s} \in \mathcal{M}_{en}(t_j, q)} k'_s$.
- Step 5: If $\tau_q \tau_k < l_j$ then jump to Step 9.
- Step 6: If $\tau_q \tau_k > u_j$ then jump to Step 9.
- Step 7: $T_q^n \leftarrow T_q^n \bigcup t_j$.
- Step 8: Jump to Step 10.
- Step 9: $T_q^{un} \leftarrow T_q^{un} \bigcup t_j$.
- Step 10: End.

Algorithm 3: Testing of a missing firing occurrence.

For each transition $t_j \notin T_q$

- Step 1: Let τ_{prev_j} be the time of the last firing of t_j before τ_q : if such a time does not exist, i.e., t_j fires for the first time at τ_q , then $\tau_{prev_j} := 0$.
- Step 2: If t_j belongs to a choice, i.e., $t_j \in p^{\bullet}$ with $|p^{\bullet}| > 1$, and there exists $t_i \in p^{\bullet}$ such that the marking m_{k_s} has enabled it at time τ_k and $t_i \in T_q$, then, if $u_j \ge \delta(t_i, k, q)$, jump to Step 8.
- Step 3: Collect in the set named $\mathcal{M}_{en}(t_j, q)$, each marking $m_{k'_s}$ reached at a time $\tau_{k'}$ such that $\tau_{k'} \geq \tau_{prev_j}$ and $\tau_{k'}$: $\tau_q \tau_{k'} \leq l_j$, for which condition (2) holds and $\nexists \tau_{k''_s}$: $\tau_{k'_s} < \tau_{k''_s} < \tau_q$ s.t. $m_{k''_s}$ does not enable t_j . The markings in the set $\mathcal{M}_{en}(t_j, q)$ are candidate markings for the enabling of $t_j \notin T_q$.
- Step 4: If $\mathcal{M}_{en}(t_j, q) = \emptyset$ then jump to Step 8.



Fig. 6. Evolution of the net system of Example 3.

Step 5: Choose m_{k_s} as the oldest marking of $\mathcal{M}_{en}(t_j, q)$, thus $m_{k_s} \in \mathcal{M}_{en}(t_j, q)$ and $k_s = \min_{\forall k'_s \text{ s.t. }} m_{k'_s} \in \mathcal{M}_{en}(t_j, q) k'_s$. Step 6: If $\tau_k + u_j > \tau_q$ then jump to Step 8. Step 7: $T_q^{miss} \leftarrow T_q^{miss} \bigcup t_j$. Step 8: End.

Remark 1: At Step 4 (Step 5) of Algorithm 2 (Algorithm 3), in case the cardinality of the set $\mathcal{M}_{en}(t_j, q)$ is greater than 1, i.e., there is more than one candidate enabling marking for the firing of transition t_j , the oldest one is always chosen as the one that has enabled the firing of t_j . The reason is that TPN semantic forces an enabled transition t_j to fire not before a time equal to its lower bound is spent from its enabling and within a time equal to its upper bound is elapsed from its enabling. If a *younger* marking is selected as the enabling one, a shorter elapsed time from the enabling is wrongly considered for identification purposes.

The following example better clarifies how the occurrence of an anomalous behavior is tested.

Example 3: Consider the system of Fig. 3(b) and the observed timed firing sequence $\mathfrak{S} = \mathfrak{S}_{prev}(T_q, \tau_q)$ such that $\mathfrak{S}_{prev} = (\{t_{11}\}, 0) \ (\{t_1, t_3\}, 1)$ and $(T_q, \tau_q) = (T_3, \tau_3) = (\{t_2, t_4\}, 1.3)$. To test if an unexpected firing has occurred at time $\tau_3 = 1.3$, it is necessary to verify if t_2 and t_4 belong to T_3^{un} so Algorithm 2 is executed: since both transitions fire for the first time at τ_3 , $\tau_{prev_2} = 0$ and $\tau_{prev_4} = 0$; the set $\mathcal{M}_{en}(t_2, 3)$ is composed by the markings m_1 , shown in Fig. 6(a), and m_2 , Fig. 6(b), since both markings satisfy condition (2), have been reached, respectively, at the time τ_1 and τ_2 greater than τ_{prev_2} and that $\tau_3 - l_2 = 0.3$, since $l_2 = 1$. Since step 1 precedes step 2, m_1 is selected as the enabling marking of the firing of t_2 at time τ_3 , consequently $\tau_k = \tau_1 = 0$. Since $\tau_q - \tau_k = 1.3 - 0 = 1.3 = l_2, t_2 \in T_3^n$, thus no unexpected firing of t_2 has occurred at time τ_3 .

The algorithm is executed again for testing the firing of t_4 : since $\mathcal{M}_{en}(t_4, 3) = \emptyset$ (as it can be verified looking to Fig. 6), an unexpected firing of t_4 has occurred at time τ_3 .

To verify if some missing firings has occurred, the Algorithm 3 is executed for each transition that does not belong to T_3 . From Fig. 6 it is simple to verify that at time τ_3 , t_5 is the only other enabled transition besides t_2 and t_4 , and it has been enabled at τ_2 . Since $\tau_q - \tau_k = 1.3 - 1 = 0.3 < l_5$ and for all the other transitions $\mathcal{M}_{en}(t_j, 3) = \emptyset$, no missing firings occurred.

Algorithm 2 and 3 are used to detect an unexpected firing and/or a missing firing. However, in practice, any tool capable to verify the correctness of a TPN behavior can be used to detect the occurrence of a fault.

C. Identification of the repaired model

The repaired model of the system, named \tilde{S} , is obtained on the basis of S_0 , \mathfrak{S} and \mathcal{L}_{obs}^F , solving the following algebraical linear system:

$$G(S_0, \mathfrak{S}, \mathcal{L}^F_{obs}) = \bigcup_{\forall \overline{\mathfrak{S}} \in \mathcal{L}^F_{obs} \bigcup \mathfrak{S}} G_A(T^{un}_q, T^{miss}_q, q)$$
(3)

Linear constraints $G_A(T_q^{un}, T_q^{miss}, q)$ are obtained starting from logical conditions that will be presented in Section V and applying the transformation rules presented in [17] (an example of logical condition transformation and linearization is presented in Section V-C and Section V-D, respectively).

From now on, with an abuse of notation $G_A(T_q^{un}, T_q^{miss}, q)$ (and $G_I(T_q^{un}, q)$ and $G_F(T_q^{un}, T_q^{miss}, q)$ as well, introduced in the following) is used to indicate both the logical conditions as well as the linear system into which they can be transformed.

Since, in general, the solution of $G(S_0, \mathfrak{S}, \mathcal{L}_{obs}^F)$ is not unique, to select one among these solutions a performance index is given and, solving an appropriate MILPP, a TPN system that minimizes the considered performance index is determined.

Moreover, modifying the objective function it is possible to prefer the extension of the firing interval or the adding of fault transitions as possible explanation of the faulty behavior.

In particular if $f(\operatorname{Pre}_f, \operatorname{Post}_f, \Delta \mathbf{l}, \Delta \mathbf{u})$ is the considered performance index, where $\Delta \mathbf{l}, \Delta \mathbf{u} \in Q^{+n}$ are, respectively, the vectors of the extension of the lower and upper bounds firing times of the nominal transitions then an identification problem can be formally stated as follows

$$\min_{s.t. \ G(S_0,\mathfrak{S},\mathcal{L}_{obs}^F)} \ f(\mathbf{Pre}_f,\mathbf{Post}_f,\mathbf{\Delta l},\mathbf{\Delta u})$$
(4)

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

$$f(\mathbf{Pre}_{f}, \mathbf{Post}_{f}, \Delta \mathbf{u}) = \sum_{i=1}^{m} \sum_{j=1}^{n_{f}} a_{ij} (\mathbf{Pre}_{f}(p_{i}, t_{f_{j}}) + \mathbf{Post}_{f}(p_{i}, t_{f_{j}})) + \sum_{i=1}^{n} b_{i} (\frac{\Delta l_{i}}{l_{i}} + \frac{\Delta u_{i}}{u_{i}}),$$
(5)

opportunely balancing the value of each a_{ij} and b_i , it is possible to find a trade-off between the minimization of the sum of the arc weights of the fault transitions and of the relative extension of the firing interval for each nominal transition. The key points are some *a priori* knowledge of the system and/or some considerations on its layout.

An arc going from a fault transition to a place of the nominal model means that the marking of this place can be modified/repaired. In particular, a high weight associated with such an arc in the performance index is equivalent to assume a low probability of having it in the faulty system. Thanks to the local state representation of PN models, it is reasonable to assume that the meaning of a place and, consequently, the probability that its marking can be modified by a fault transition, are known, as also shown in the case study.

The same occurs with the firing interval extension. It is reasonable assuming to be known if the duration of a certain activity can be affected or not by an extension, and, consequently, the probability to have a firing interval extension since each activity is associated to a transition.

Moreover, in several real applications it is known *a priori* that a fault that may affect a given subnet, has no effect on some parts of the net. In such a case, it is sufficient to impose that some entries in the \mathbf{Pre}_f and \mathbf{Post}_f matrices are null, thus reducing the number of unknowns.

On the same level, in several real applications it is known *a* priori that the duration of some activities may be not affected by a fault. In such a case, it is sufficient to impose that some entries in the Δl and Δu vectors are null, so the number of unknowns as well as of constraint is reduced.

For example, while it reasonable to assume that the time a conveyor belt takes to carry an item from its input point until its output point can changes from the expected one because of the wear of the motor and/or of the transmission belt, assuming this for the time a controller takes to make a decision, is not.

Moreover, when a time-out occurs, the proposed algorithm consider transitions with a null Δu entry.

In the case study, it is shown how, in practice, significant reduction of computation time up can be achieved reducing the unknowns, as discussed above.

Finally, when the optimization problem is solved, an assumption about the value of the number of fault transitions of the repaired model of the system must be done. Such a number is denoted by n_f . An approach is to assign a starting value to n_f (e.g. $n_f = \overline{n_f}$) and try to solve the system of equations: if it gives no solutions, n_f is incremented. On the other hand, if a solution is found, n_f can be progressively reduced to obtain a more compact model, until no solution is found.

V. LOGICAL CONDITION FORMULATION

Unexpected firings as well as missing firings of transitions at time τ_q can be model both with a firing interval extension and with the adding of fault transitions to the nominal model of the system.

As a consequence, given the faulty behavior of the system at time τ_q , $(T_q^{un}, T_q^{miss}, q)$, the system that exhibits such a behavior satisfies the following logical condition:

 $G_A(T_q^{un}, T_q^{miss}, q)$:

$$G_I(T_a^{un}, q) \bigvee G_F(T_a^{un}, T_a^{miss}, q) \tag{6}$$

Condition $G_I(T_q^{un}, q)$ holds when the observed anomaly can be modeled as a firing time extension, whereas condition



Fig. 7. Evolution of the system of Fig. 3: marking reached after the firing of $\mathfrak{S}_{prev} = (\{t_{11}\}, 0)(\{t_2, t_4\}, 1)(\{t_1\}, 1.2)$

 $G_F(T_q^{un}, T_q^{miss}, q)$ holds when the anomaly can be modeled by adding some fault transitions.

In the next sections it will be shown how these conditions are obtained.

A. Logical conditions formulation for the extension of the transition firing interval

Let S be the current model of the system that generates the language $\mathcal{L}(S)$ and $S_0 = \langle N, \mathbf{m}_0, I \rangle$ be the nominal model of the system, with $N = (P, T, \mathbf{Pre}, \mathbf{Post})$.

For each timed firing sequence $\mathfrak{S} = \mathfrak{S}_{prev}(T_q, \tau_q)$ such that:

• $\mathfrak{S} \notin \mathcal{L}(S);$

• $\mathfrak{S}_{prev} \in \mathcal{L}(S);$

• \mathfrak{S}_{prev} is a subsequence of \mathfrak{S} , of length q-1;

the problem is to determine the extended firing interval I^{ex} , such that $I^{ex}(t_j) = [l_j - \Delta l_j, u_j + \Delta u_j]$, where Δl_j and Δu_j are positive rational numbers, l_j and u_j are, respectively, the lower and upper bound of the nominal firing interval of transition t_j , in the way that the resulting system $\tilde{S} =$ $\langle N, m_0, I^{ex} \rangle$ generates the language $\mathcal{L}(\tilde{S})$ that includes \mathfrak{S} .

Notice that $\mathcal{L}(S) \supseteq \mathcal{L}(S_0)$ since the net structure is the same and transition firing intervals have been extended.

Proposition 1 (Unexpected firing of t_j at τ_q): Let t_j be a transition belonging to the set T_q^{un} , enabled at time τ_k by the marking m_{k_s} . The unexpected firing of t_j at τ_q is modeled by an extension of the firing interval of t_j if there exists a value $\Delta l_j \in Q^+$ or a value $\Delta u_j \in Q^+$ for which the logical condition named $G_{un}(t_j, q, k)$ holds, with

 $G_{un}(t_j,q,k):$

$$\Delta l_j \ge \tau_k - \tau_q + l_j \bigvee \Delta u_j \ge \tau_q - \tau_k - u_j. \tag{7}$$

Proof: Since an enabled timed transition must fire in a time belonging to $I(t_j)$ from its enabling, condition (7) imposes that an extension Δl_j has occurred to explain the firing of t_j in a time less than l_j or it imposes that an extension Δu_j has occurred to explain the missing firing of t_j at the time $\tau_q > \tau_k + u_j$.

Proposition 1 is extended to the whole set T_q^{un} by means of the logical condition named $G_I(T_q^{un}, q)$, with

 $G_I(T_q^{un},q):$

$$\bigwedge_{\forall t_j \in T_a^{un}} G_{un}(t_j, q, k) \tag{8}$$

Example 4: Consider the system of Fig. 3(b) and the observed sequence $\mathfrak{S} = \mathfrak{S}_{prev}$ (\emptyset , 2) = ({ t_{11} }, 0) ({ t_2 , t_4 }, 1) ({ t_1 }, 1.2) ({ t_6 }, 1.8). The marking reached after the firing of \mathfrak{S}_{prev} is shown in Fig. 7. At this marking all the transitions are disabled except t_6 .

At time $\tau_4 = 1.8$ an unexpected firing of t_6 is observed; consequently $T_4^{un} = \{t_6\}$ and $T_4^{miss} = \{\emptyset\}$.

It is simple to verify that the firing interval I^{ex} such that

$$I^{ex}(t_j) = \begin{cases} [l_6 - 0.2, u_6] & \text{if } j = 6\\ I(t_j) & \forall j \neq 6 \end{cases}$$

 \Diamond

justify the faulty behavior of the system.

B. Logical conditions formulation for the adding of fault transitions

Let S be the current model of the system, that generates the language $\mathcal{L}(S)$ and $S_0 = \langle N, \mathbf{m}_0, I \rangle$ be the nominal model of the system, with $N = (P, T, \mathbf{Pre}, \mathbf{Post})$. For each timed firing sequence $\mathfrak{S} = \mathfrak{S}_{prev}(T_q, \tau_q)$ such that:

- $\mathfrak{S} \notin \mathcal{L};$
- $\mathfrak{S}_{prev} \in \mathcal{L};$
- \mathfrak{S}_{prev} is a subsequence of \mathfrak{S} , of length q-1;

given the set of fault transition T^f , with cardinality n_f , the goal is to identify the faulty incidence matrices \mathbf{Pre}_f and \mathbf{Post}_f , with dimension $m \times n_f$ such that, \mathfrak{S} belongs to the language generated by $\tilde{S} = \langle \tilde{N}, \mathbf{m}_0, \tilde{I} \rangle$, where $\tilde{N} = (P, T \bigcup T^f, [\mathbf{Pre} \ \mathbf{Pre}_f], [\mathbf{Post} \ \mathbf{Post}_f])$ and

$$\tilde{I}(t) = \begin{cases} I(t) & \text{if } t \in T\\ [0, \infty[\cap \mathcal{Q}^+] & \text{if } t \in T^f \end{cases}$$

The firing interval of the nominal transitions does not change in \tilde{S} , while the firing interval of the fault transitions is set equal to $[0, \infty[$. This setting arises from the consideration that fault transitions are associated to events caused by the wear as well as by unpredictable failures.

Notice that $\mathcal{L}(S) \supseteq \mathcal{L}(S_0)$ since S is obtained without any modification of arcs, transitions or places of the nominal model and the firing interval of the fault transitions, being set equal to $[0, \infty]$, does not forbid any sequence of the nominal language.

Just to not forbid any sequence of the nominal language, firing of fault transitions must not modify the enabling of those transitions whose behavior is coherent with the nominal model. This motivates the following assumption.

Assumption 3: Firings of fault transitions lead to a marking that enables the unexpected firings of transitions in T_q^{un} and disables the firing of transitions belonging to T_q^{miss} but they never forbid sequences that are enabled in the nominal behavior.

Assumption 4: The subnet induced by the fault transition is acyclic. \diamond

Assumption 4 is justified by the fact that the proposed approach is based on the incidence matrix and on the state equation of the net. As a consequence of Assumption 4, all fault transitions are loop-free so the incidence matrix contains all the information on the net structure. Moreover, it guarantees necessary and sufficient conditions for reachability in the unobservable subnet.

In the following the logical conditions to satisfy for identifying the system \tilde{S} are presented.

Proposition 2 (Unexpected firing of t_j at τ_q): The unexpected firing of $t_j \in T_q^{un}$ at time τ_q has been enabled by the firings of at least one fault transition $t_{f_h} \in T^f$ if the following logical condition holds:

 $G_{unexpected}(t_j, q)$:

$$m_{q-1} + \sum_{h=1}^{n_f} \alpha_{hjq} (\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})) \ge \mathbf{Pre}(\cdot, t_j) + \sum_{\forall t_i \in T_{q-1}^{en}} \mathbf{Pre}(\cdot, t_i),$$
(9)

where α_{hjq} is a boolean variable such that $\alpha_{hjq} = 1$ if t_{f_h} has fired enabling the firing of t_j at time τ_q , otherwise $\alpha_{hjq} = 0$, and T_{q-1}^{en} is the set made up of transitions that, enabled at τ_{q-1} , continue to be also enabled at τ_q \diamond

Proof: Condition (9) imposes that marking m_k , reached starting from m_{q-1} after the firing of at least one fault transition t_{f_h} , is such that $m_k \geq \operatorname{Pre}(\cdot, t_j)$. Moreover, it imposes that m_k enables transitions in T_{q-1}^{en} too.

Proposition 3 (Missing firing of t_i): The firing of transition $t_i \in T_q^{miss}$, enabled at a marking reached at time τ_z , is disabled at time τ_q by the firing of at least one fault transition $t_{f_h} \in T^f$ if the following logical condition holds: $G_{missing}(t_i, q)$:

$$\underbrace{\mathbf{m}_{q-1} + \sum_{h=1}^{n_f} \alpha_{hiq}(\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})) < \mathbf{Pre}(\cdot, t_i)}_{(10.1)}}_{\mathbf{m}_{q-1} + \sum_{h=1}^{n_f} \alpha_{hiq}(\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})) \geq \sum_{\forall t_i \in T_{q-1}^{en}} \mathbf{Pre}(\cdot, t_i)}_{\forall t_i \in T_{q-1}^{en}} \underbrace{(10)}_{(10.2)}$$

where α_{hiq} is a boolean variable such that $\alpha_{hiq} = 1$ if t_{f_h} has fired disabling the firing of t_i at time τ_q , otherwise $\alpha_{hiq} = 0$.

Proof: Condition (10.1) imposes that the marking reached starting from m_{q-1} after the firing of at least one fault transition disables the firing of t_i ; condition (10.2) imposes that the firing of t_{f_h} does not affect the enabling of transitions in T_{q-1}^{en} .

Lemma 1: The unobservable firing of a fault transition $t_{f_h} \in T^f$, occurred at the time $\tau_k \leq \tau_q$ has been enabled at marking m_{r_s} , reached at time $\tau_r : \tau_r \leq \tau_q$, if the following logical condition holds:

 $G_{fen}(t_{f_h}, r, s):$

$$\boldsymbol{m}_{r_s} \ge \mathbf{Pre}_f(\cdot, t_{f_h}) \tag{11}$$

Proof: Condition (11) imposes that marking m_{r_s} enables the firing of t_{f_h} .

Proposition 4 (Faulty Behavior): The system $\tilde{S} = \langle \tilde{N}, m_0, \tilde{I} \rangle$ justifies the faulty behavior of the system at time τ_q , if the following logical condition holds:

 $G_F(T_q^{un}, T_q^{miss}, q):$

$$\underbrace{\bigwedge_{\forall t_j \in T_q^{un}} G_{unexpected}(t_j, q)}_{(12.1)} \underbrace{\bigwedge_{\forall t_i \in T_q^{miss}} G_{missing}(t_i, q)}_{(12.2)}}_{\text{IF} (\alpha_{hjq} = 1 \bigvee \alpha_{hiq} = 1)} \text{THEN} \begin{array}{c} \text{i} \\ \text{i} \\ \mathcal{V}_{\forall (r,s) \in \Gamma} G_{fen}(t_{f_h}, r, s) \\ \text{i} \\ \mathcal{V}_{\forall (r,s) \in \Gamma} \bar{z}_{hqrs} = 1 \end{array}$$

$$\bigwedge \forall t_{f_h} \in T^f, \ \forall x \in [1,m] \\ \nexists p_x : \mathbf{Pre}_f(p_x, t_{f_h}) > 0 \bigwedge \mathbf{Post}_f(p_x, t_{f_h}) > 0$$



Fig. 8. Faulty system \tilde{S} admitting sequence: (a) $\mathfrak{S} = \mathfrak{S}_{prev}(\emptyset, 2) = (\{t_{11}\}, 0)(\{t_2, t_4\}, 1)(\{t_1\}, 1.2)(\emptyset, 2);$ (b) $\mathfrak{S} = \mathfrak{S}_{prev}(\{t_2, t_4\}, 1.3) = (\{t_{11}\}, 0)(\{t_1, t_3\}, 1)(\{t_2, t_4\}, 1.3).$

where Γ is the set of couple of indexes (r, s) that characterize any substep r_s preceding the step q of the sequence \mathfrak{S} , thus any substep r_s reached at the time $\tau_r \leq \tau_q$, and \overline{z}_{hqrs} is a boolean variable equal to 1 when the marking m_{r_s} enables the firing of t_{f_h} , i.e., it is equal to 1 when the logical condition $G_{fen}(t_{f_h}, r, s)$ holds. \Diamond

Proof: Logical conditions (12.1) and (12.2) respectively extend to the set T_q^{un} and T_q^{miss} conditions of Proposition 2 and Proposition 3.

Logical condition (12.3) regards fired fault transitions; it imposes that: i) the firing of each transition t_{f_h} is enabled at a marking m_{r_s} reached at a time $\tau_r \leq \tau_q$; ii) if more than one marking is candidate for the enabling of the firing of t_{f_h} at τ_k , i.e., there exists more than one marking for which condition (11) is satisfied, then just one marking m_{r_s} is selected as the enabling marking of the firing.

Finally logical condition (12.4) imposes that each fault transition is loop-free in accord to Assumption 4.

Example 5: Consider again the system of Example 1 and the observed sequence $\mathfrak{S} = \mathfrak{S}_{prev}(\emptyset, 2) = (\{t_{11}\}, 0)$ $(\{t_2, t_4\}, 1) \ (\{t_1\}, 1.2) \ (\emptyset, 2)$

It is simple to verify that the system of Fig. 8(a) satisfies condition (12). Indeed t_{f_1} , firing at time $\tau_k \in [1, 2]$, disables the firing of t_6 while it does not affect the enabling of any other transition, satisfying condition (12.2); it is enabled at marking m_2 reached at time $\tau_2 = 1$ and hence conditions i) and ii) of (12.3) hold. Finally also condition (12.4) is satisfied since there are no self loops involving t_{f_1} . Consequently (12) holds. \Diamond

Example 6: Consider the same system of Example 5 and the observed sequence $\mathfrak{S}' = \mathfrak{S}'_{prev}$ ($\{t_2, t_4\}, 1.3$) = ($\{t_{11}\}, 0$) ($\{t_1, t_3\}, 1$) ($\{t_2, t_4\}, 1.3$). As shown in the Example 3, at τ_3 an unexpected firing of t_4 is observed, consequently $T_3^{un} = \{t_4\}$ and $T_3^{miss} = \emptyset$.

The system of Fig. 8(b) admits the observed faulty behavior. Indeed t_{f_1} , firing at time $\tau_k \in [0, \tau_3]$, enables the unexpected firing of t_4 and it neither disables the firing of t_5 nor enables any other transition, satisfying condition (12.1); it is enabled at marking m_0 and hence conditions i) and ii) of (12.3) hold. Finally, also condition (12.4) is satisfied since there are no self loops involving t_{f_1} . Consequently (12) holds since no missing firings are occurred. The same problem has no solution when firing interval extension technique is adopted.

C. Transformation of logical conditions into linear constraints

Applying the rules presented in [17], logical conditions introduced in Section V can be rewritten as sets of linear constraints. As an example, in the following it is shown the transformation of logical condition (12.3) into the set (13). $\forall t_{f_b} \in T^f$:

$$\begin{array}{c} z_{hqrs} + \bar{z}_{hqrs} = 1; \\ z_{hqrs}, \bar{z}_{hqrs} \in \{0, 1\}; \\ \lambda_{hjq} + \alpha_{hiq} + K z_{en_q} \ge 1; \\ \alpha_{hjq} + \alpha_{hiq} + K \bar{z}_{en_q} \ge 0; \\ \alpha_{hjq} + \alpha_{hiq} - K \bar{z}_{en_q} \le 0; \\ z_{en_q} + \bar{z}_{en_q} = 1; \\ z_{en_q}, \bar{z}_{en_q} \in \{0, 1\}; \\ K \in \mathbb{N}; K > 1; \\ \mathbf{Pre}_{\mathbf{f}}(\cdot, t_{f_h}) - K_{1} z_{en_q} - K_{1} z_{hqrs} \le \mathbf{m}_{r_s}; \\ \mathbf{K}_{1} \in \mathbb{N}^{m}; \mathbf{K}_{1} > \mathbf{m}_{r_s}; \\ \Sigma_{\forall (r,s) \in \Gamma} \bar{z}_{hqrs} + K_{3} z_{en_q} \ge 1; \\ \Sigma_{\forall (r,s) \in \Gamma} \bar{z}_{hqrs} - K_{3} z_{en_q} \ge 1; \\ \Sigma_{\forall (r,s) \in \Gamma} \bar{z}_{hqrs} - K_{3} z_{en_q} \le 1; \\ K_{3} \in \mathbb{N}; K_{3} > 1; \end{array} \right\} (13.2)$$

Constraints (13.0) introduce the dummy boolean variables z_{hqrs} needed to the transformation of the logical conditions into linear constraints; constraints (13.1) impose that the boolean variable z_{enq} is equal to zero when the IF tested condition holds; constraints (13.2) and (13.3) correspond, respectively, to conditions i) and ii).

D. Constraints linearization

As shown in [8], the nonlinearity of condition (9.1), due to the product of α_{hjq} and $\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})$, is removed rewriting (9.1) as the following set of linear constraints

$$\begin{array}{c} \boldsymbol{m}_{q-1} + \sum_{h=1}^{n_f} \boldsymbol{c}_{hjq} \geq \\ \mathbf{Pre}(\cdot, t_j) + \sum_{\forall t_i \in T_{q-1}^{en}} \mathbf{Pre}(\cdot, t_i); \\ \boldsymbol{c}_{hjq} - v(\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})) - \\ + \boldsymbol{K}_5 z_{hjq_v} \geq \mathbf{0}^m; \\ \boldsymbol{c}_{hjq} - v(\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})) - \\ + \boldsymbol{K}_5 z_{hjq_v} \leq \mathbf{0}^m; \\ z_{hjq_0} + z_{hjq_1} = 1; \\ z_{hjq_0}, z_{hjq_1} \in \{0, 1\}; \end{array} \right\} \quad \forall v \in \{0, 1\},$$

$$(14)$$

with $K_5 = K_5 \mathbf{1}^m$ such that K_5 is a very large constant that ensures the relation $K_5 > c_{hjq} - v(\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h}))$ holds.

For each fault transition t_{f_h} , a vector c_{hjq} is introduced that substituted the unlinear product of α_{hjq} and $\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})$; c_{hjq} is equal to $\mathbf{0}^m$, when $z_{hjq_0} = 0$ or it is equal to $\mathbf{Post}_f(\cdot, t_{f_h}) - \mathbf{Pre}_f(\cdot, t_{f_h})$, when $z_{hjq_1} = 0$. In the first (second) case, the first constraint of (14) correspond to logical condition (9.1) when $\alpha_{hjq} = 0$ ($\alpha_{hjq} = 1$).

The same procedure can be also applied to linearize logical condition (10).

VI. CASE STUDY

The approach presented in this paper are based only on Assumptions 1 and 2 discussed in Section II-B, no assumptions are required about the net topology. Then, it can be successfully applied especially to Flexible Manufacturing Systems and Workflow Management Systems, where the single server semantic and the enabling memory policy are not so relevant as they are in Computer and Communication systems.

In this section, the proposed approach is used to identify the repaired model of the automatic material handling system prototype installed at the Automatic Control and Robotics Laboratory of the University of Salerno, shown in Fig. 9(a), but it is very general and can be applied directly in many contexts, as for example automated warehouses [28] and multi-robot systems [29].

The proptotyle is a modular system, made up of conveyor belts and elevators that move stock units in horizontal as well as in vertical way. The handling system is subdivided in three zones named, respectively, Zone A, Zone B and Zone C and each zone is subdivided in four level, (Level 1- Level 4 in Fig. 9(b)).

The system is controlled by means of a PLC Siemens S300 able to acquire sensors values and command actuators by means of three remote I/O modules, each one associated to a different zone, which communicate with the PLC by means of a field bus.

The handling system can be used to execute several handling tasks, one of this is hereinafter described: an item is introduced into the handling system from the input point (belt A1 in Fig. 9(b)); it is moved on the elevator AL and sequentially is moved up to Level 2 and 3. At the *i*-th level the item stays for a time $T_i \in [T_{i_{min}}, T_{i_{max}}]$ to allow that unspecified operations can be executed on it. After this time it is moved to the following level. Finally it is moved to Level 4 and after a time $T_4 \in [T_{4_{min}}, T_{4_{max}}]$ it is moved to Zone C, toward the belts B4-B11. When it arrives at the end of B11, the elevator CL goes up to Level 4 and moves the item down to Level 1. Then the item passes on belt C1 and finally it goes out from the system.

The TPN model in Fig. 9(c), without considering transition t_{f1} and the red dotted arcs, has been assumed as the nominal model of the task; it is made up of 37 transitions and 42 places and represents the task as it has been programmed in the system; the duration of its timed activities has been obtained from measures. For the sake of clarity, transition firing intervals are not reported in the figure but are indicated in Table III together with the event associated to each transition. For the same reason, only significative places have been labeled: their meaning is reported in Table IV.

Firing interval $[0, \infty[$ has been assigned to i) transitions associated to a manual operation that can be executed at any time by a human worker (as in the case of transition t_1 and t_{22} , which fires when an item is put in and taken out from the system); ii) controlled transitions, since, even if they are enabled under the current marking, they fires only if the PLC does not disable their firing.

The marking shown in Fig. 9(c) has been assumed as the initial condition of the system: it is such that all the levels, as well as the input and output points, are empty and both the elevators are free and placed at Level 1.

According to the algorithm presented in Section IV, the faulty behavior $\mathcal{L}_{obs}^{F} = \{\mathfrak{S}', \mathfrak{S}''\}$ with

$$\mathfrak{S}' = \mathfrak{S}'_{prev}(\{t_7\}, 60.01) = (\{t_1\}, 5.01) (\{t_2, t_3\}, 15.01)$$







Fig. 9. (a) The prototype of automated handling system installed at the University of Salerno; (b) its layout; (c) TPN nominal model of the case study.

TABLE III MEANING OF TRANSITIONS OF THE CASE STUDY.

| Transition | Firing Interval (s) | Event |
|----------------------------------|---|-------------------------|
| t_1 | $[0,\infty[$ | A new item at the in- |
| | | put point. |
| t_2 | [6,10] | Item on A1. |
| $t_3 (t_8, t_{13})$ | $[0,\infty[([0,\infty[, [0,\infty[)$ | Start transfer from A1 |
| | | (B2, B3) to AL. |
| $t_4 \ (t_9, t_{14})$ | [7.75,13.75] | Item from A1 (B2, B3) |
| | ([14.67,16.67], | on AL. |
| | [9.66,16.66]) | |
| $t_5 (t_{10}, t_{15})$ | [17.01 ,19.01] ([17.01 | Busy AL arrived at |
| | ,19.01], [17.01 | Level 2 (3, 4). |
| | ,19.01]) | |
| $t_6 (t_{11}, t_{16})$ | [8.49, 15.49] ([15,17], | Item on B2 (B3, B4). |
| | [8,10]) | |
| $t_7 (t_{12}, t_{17})$ | [118,186] ([91,181], | Operation completed |
| , | [150,162]) | at Level 2 (3, 4). |
| t_{23} | [29.24,31.24] | Item on B11. |
| t_{18} | $[0,\infty[$ | Start transfer from B11 |
| , | [5 75 10 75] | to CL. |
| t_{19} | [5./5,13./5] | Item from B11 on CL. |
| t_{20} | [39.30,48.30] | Busy CL arrived at |
| <i>t</i> | [14 75 22 75] | Level 1. Item on C1 |
| <i>L</i> 21 | [14.75, 22.75] | Broads completed |
| t22 | $[0,\infty]$ | Empty CL starts to go |
| 137 | [0,∞[| up to Level 4 |
| tac | [49 00 51 00 1 | Empty CL arrived at |
| 630 | [49.00,91.00] | Level 4 |
| t_{24} (too t_{22}) | $(1 \propto 0]$ $(1 \propto 0]$ $(1 \propto 0]$ | Empty AL starts to go |
| 024 (028, 033) | | down to Level 3 (2, 1) |
| t_{25} (t_{20} , t_{24}) | [8,74,16,74] | Empty AL arrived |
| 023 (029, 034) | ([14.76.16.76]. | down at Level 3 (2. |
| | [14.75.16.75]) | 1). |
| t_{27} (t_{31}, t_{35}) | $[0,\infty[([0,\infty[, [0,\infty[)$ | Empty AL starts to go |
| 2. (31/ 00/ | | up to Level 4 (3, 2). |
| t_{26} (t_{30}, t_{32}) | [8.74,16.74] | Empty AL arrived up |
| 20 (00/ 02/ | ([12.00,19.00], | at Level 3 (2, 1). |
| | [14.75,16.75]) | × · · / |

TABLE IV Meaning of the principal places of the case study.

| Place | Meaning |
|-------------------------------------|--------------------------------|
| $p_1 (p_5)$ | Input (Output) point is empty. |
| $p_2 (p_3, p_4)$ | Level 2 (3, 4) is empty. |
| $p_6 (p_7, p_8, p_9)$ | Busy AL at Level 1 (2, 3, 4). |
| $p_{10} (p_{11})$ | Busy CL at Level 4 (1). |
| $p_{12}(p_{13})$ | Empty CL at Level 1 (4). |
| p_{14} (p_{15}, p_{16}, p_{17}) | Empty AL at Level 1 (2, 3, 4). |

 $\begin{array}{l} (\{t_4\},28.01)\;(\{t_5\},46.01)\;(\{t_6\},55.00)\;(\{t_7\},60.01);\\ \mathfrak{S}''\;=\;\mathfrak{S}''_{prev}(\{t_{16}\},93.01)=\;(\{t_1\},12.00)\;(\{t_2,t_3\},18.00)\;\\(\{t_4\},26.00)\;(\{t_5\},44.01)\;(\{t_6\},59.00)\;(\{t_{16}\},93.01);\\ \text{has been observed. Moreover, the current faulty sequence}\\ \mathfrak{S}\;=\;\mathfrak{S}_{prev}(\{t_{11}\},299.03)=\;(\{t_1\},6.01)\;(\{t_2,t_3\},16.01)\;\\(\{t_4\},29.00)\;\;(\{t_5\},47.01)\;\;(\{t_6\},56.01)\;\;(\{t_7,t_8\},242.01)\;\\(\{t_9\},258.00)\;\;(\{t_{10}\},276.01)\;\;(\emptyset,293.01)\;\;(\{t_{11}\},299.03),\\ \text{has been observed too.} \end{array}$

Notice that \mathfrak{S}_{prev} is itself a faulty sequence, since a timeout occurred at step 9. However, the criticality of such a fault has been considered low, and consequently observation has not been stopped after the identification of the faulty model.

The identification problem $G(S_0, \mathfrak{S}, \mathcal{L}^F_{obs})$, with respect to the nominal model S_0 , the current observation \mathfrak{S} and the faulty observed language \mathcal{L}^F_{obs} , when the objective function is the

one in (5) with $a_{ij} = 1 \ \forall i = [1, ..., m], \forall j = [1, ..., n_f]$ and $b_i = 1 \ \forall i = [1, \dots, n]$, and $n_f = 3$, has been solved by Cplex[©] running on a PC equipped with Intel[©] Core[™] i7 CPU at 2.67 GHz, 8.00 GB of Ram and a 64 bit operative system; the computation time takes 39.78 seconds.

The observed faulty behavior has been modeled by the extension of the firing interval of transitions t_7 and t_{11} and the adding of one fault transition, t_{f_1} , drawn in red in Fig. 9, linked to the rest of the net by means of the red dotted arcs.

The occurrence of \mathfrak{S}' has been caused by a drastic short duration of the operations at Level 2, that can be due to a wrong handling of the activity by a human worker; as a consequence the lower bound of transition t_7 has been reduced of the amount $\Delta l_7 = 112.99s$. The occurrence of \mathfrak{S}'' has been caused by the omission of the carriage of the item to Level 3: soon after that the item is arrived at the end of B2, it is moved at Level 4; the firing of fault transition t_{f_1} models such a fault. Finally \mathfrak{S} is due to a longer duration of the transfer of the item from the elevator AL to the belt B3, that can be due to an accidental block of the item caused by an incorrect positioning of the item itself on the belt. Such a fault has been modeled by the increment of the upper bound of transition t_{11} of the amount $\Delta u_{11} = 6.02s$.

Remark 2: The execution of the repair model identification algorithm after the observation of the faulty timed firing sequence \mathfrak{S}_{prev} would return an intermediate repaired model, where the missing firing of transition t_{11} is modeled by the adding of a second fault transition t_{f_2} with preset $\bullet t_{f_2} = \{p_8\}$ and with postset $t_{f_2}^{\bullet} = \{\emptyset\}$ to the nominal model, without enlarging the firing interval of t_{11} ; for the sake of brevity, this intermediate repaired model is not shown. When the sequence \mathfrak{S} is observed, the firing of t_{11} at step 10 ($\tau_{10} = 299.03$) produces an unexpected firing since the intermediate model can not explain such a firing. Consequently a new repaired model is obtained where the upper bound u_{11} is incremented of the amount $\Delta u_{11} = 299.03 - 293.01 = 6.02s$ and the transition t_{f_2} disappears.

A second identification problem has been carried on with some null entries in \mathbf{Pre}_f and \mathbf{Post}_f .

From an *a priori* knowledge of the system about the reliability of lifter AL, it is reasonable to assume that fault occurrences do not interest the lifter movements. Consequently, it is assumed that none fault transition can be connected to places of the set $P_{lifter} = \{14, 37, 38, 39, 40, 41, 42\}$, made of places belonging to the subnet modeling the lifter movements.

Moreover it is also reasonable to assume that the occurrence of a fault can affect only two adjacent levels.

On the basis of this assumption, other four subsets of Phave been built:

- $P_{level1} = \{1, 2, 6, 18, 19, 20, 21, 22, 23, 24\}$ (places belonging to the part of net modeling Level 1 of the MHS);
- $P_{level2} = \{7, 24, 25, 26, 27, 28, 29, 30\}$ (places belonging to the part of net modeling Level 2 of the MHS);
- $P_{level3} = \{3, 8, 9, 29, 30, 31, 32, 33, 34\}$ (places belonging to the part of net modeling Level 3 of the MHS);
- $P_{level4} = \{4, 5, 10, 11, 12, 13, 34, 35, 36\}$ (places belonging to the part of net modeling Level 4 of the MHS).

Then, it is possible to impose that

- fault transition t_{f_1} can be connected only to places belonging to P_{level3} and P_{level4} ;
- fault transition t_{f_2} can be connected only to places belonging to P_{level2} and P_{level3} ;
- fault transition t_{f_3} can be connected only to places belonging to P_{level1} and P_{level2} .

As a consequence, 154 entries in \mathbf{Pre}_f and \mathbf{Post}_f have been set null, and using the same objective function, the same model has been identified in 22.20 seconds.

Finally, a third identification problem has been carried on using the *a priori* knowledge of the system to enforce some entries in Δl and Δu vectors to be null.

In particular, since their firing interval is assumed to be $[0,\infty[$, enlarging of firing intervals of transitions associated to a manual operation or of controlled transitions is not possible; both lifters (AL and CL) are assumed to be reliable, consequently, it is assumed that duration of the activities modeled by transitions t_5 , t_{10} , t_{15} , t_{20} , t_{36} , t_{24} , t_{28} , t_{33} , t_{25} , $t_{29}, t_{34}, t_{27}, t_{31}, t_{35}, t_{26}, t_{30}, t_{32}$ cannot be affected by any fault. Moreover, it is assumed that only the operations at level 2 can be executed in a wrong way and that only the transfer from the lift to the belt at level 3 can has anomalous duration.

This set of limitations have been included to second one, by means of 68 null entries in the vectors $\Delta l(t_i)$ and $\Delta u(t_i)$. In this case, resolution of the MILPP takes 8.27 seconds and again the same model is identified.

To prefer time enlarging with respect to the adding of a fault transition, the objective function coefficient are set to $a_{ij} = 10 \ \forall i = [1, \dots, m], \forall j = [1, \dots, n_f] \text{ and } b_i = 1 \ \forall i =$ $[1, \ldots, n]$. This last identification problem returns again the same model, and the resolution time of the second MILPP reduces to 13.24 seconds.

Thus, resolution time of the MILPP can also be influenced by the choice of the objective function. However, discussions about this argument are out of the scope of this work, since the influence of the objective function on the resolution time closely depends on the kind of used solver.

VII. 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 (4).

Problem (4) can be characterized in terms of the number of constraints and unknowns that composed it, i.e., the number of constraints and unknowns of $G(S_0, \mathfrak{S}, \mathcal{L}_{obs}^F)$.

The number of constraints and unknowns of the MILPP system $G(S_0, \mathfrak{S}, \mathcal{L}_{obs}^F)$ – depends on the following parameters:

- m = number of places of the net.
- $n_f = \text{cardinality of the set } T^f$.
- $|T_q^{un}| = \text{cardinality of the set } T_q^{un}$. $|T_q^{miss}| = \text{cardinality of the set } T_q^{miss}$.
- $|\Gamma| =$ cardinality of the set Γ (i.e., the set of couple of indexes (r, s) that characterize any substep r_s reached at the time $\tau_r \leq \tau_q$).

The number of constraints of $G(S_0, \mathfrak{S}, \mathcal{L}^F_{obs})$ is given by the sum of the number of constraints of each $G_A(T_q^{un}, T_q^{miss}, q)$, which, in turn, is obtained by summing the number of constraints of $G_I(T_q^{un}, \tau_q)$ and $G_F(T_q^{un}, T_q^{miss}, q)$, plus 1 constraint due to the OR operator.

Given the timed firing sequence \mathfrak{S} , the number of constraints of $G_I(T_q^{un}, \tau_q)$ is

$$y_{G_I}(\mathfrak{S}) = 3 \cdot |T_q^{un}| \le 3 \cdot n$$

Given the timed firing sequence \mathfrak{S} , the number of constraints of $G_F(T_q^{un},T_q^{miss},q)$ is

$$y_{G_{F}}(\mathfrak{S}) = \underbrace{[m + (2 \cdot m + 1) \cdot 2 \cdot n_{f}] \cdot |T_{q}^{un}|}_{(\text{constr. of 12.1})} + \underbrace{\{2 \cdot [m + (2 \cdot m + 1) \cdot 2 \cdot n_{f}] + 1\} \cdot |T_{q}^{miss}|}_{(\text{constr. of 12.2})} + \underbrace{[7 + (m + 1) \cdot |\Gamma|] \cdot n_{f}}_{(\text{constr. of 12.3})} + \underbrace{3 \cdot m \cdot n_{f}}_{(\text{constr. of 12.4})} \leq 2 \cdot n \cdot [m + (2 \cdot m + 1) \cdot 2 \cdot n_{f}] + n + \\ + [7 + (m + 1) \cdot (q - 1) \cdot n] \cdot n_{f} + \\ + 3 \cdot m \cdot n_{f}$$

Hence the number of constraints of $G(S_0, \mathfrak{S}, \mathcal{L}_{obs}^F)$ is

$$y_G = \sum_{\forall \overline{\mathfrak{S}} \in \mathcal{L}_{obs}^F \bigcup \mathfrak{S}} \left(1 + y_{G_I}(\overline{\mathfrak{S}}) + y_{G_F}(\overline{\mathfrak{S}}) \right)$$

The number of unknown of $G(S_0, \mathfrak{S}, \mathcal{L}^F_{obs})$ is composed of two components: a) the set of N_a unknowns and b) the set of $u^{bv}_{G(S_0,\mathfrak{S},\mathcal{L}^F_{obs})}$ boolean variables.

Consequently the total number of unknowns of $G(S_0, \mathfrak{S}, \mathcal{L}^F_{obs})$ is

$$u_{G(S_0,\mathfrak{S},\mathcal{L}_{obs}^F)} = N_a + u_{G(S_0,\mathfrak{S},\mathcal{L}_{obs}^F)}^{bv}$$
(15)

The set of unknowns consists of $2 \cdot m \cdot n_f$ integer unknowns, representing the \mathbf{Pre}_f and \mathbf{Post}_f faulty incidence matrices of the net and $2 \cdot n$ rational unknowns, representing Δl_j and Δu_j , thus the extension of the bounds of the firing interval $I(t_j)$ of each transitions of the net.

Moreover, other $6 \cdot m \cdot n_f$ integer unknowns are introduced by the linearization of the equation (9.1) and (10), for each observed faulty and missing firing; thus, given the timed firing sequence \mathfrak{S} , let $N_l(\mathfrak{S})$ be the number of unknowns due to the linearizzation of (9.1) and (10), then

$$N_l(\mathfrak{S}) = 6 \cdot m \cdot n_f \cdot (|T_q^{un}| + |T_q^{miss}|) \le 6 \cdot m \cdot n_f \cdot n$$

As a consequence

$$N_a = 2 \cdot m \cdot n_f + 2 \cdot n + \sum_{\forall \overline{\mathfrak{S}} \in \mathcal{L}_{obs}^F \bigcup \mathfrak{S}} N_l(\overline{\mathfrak{S}})$$
(16)

VIII. CONCLUSION

A Mixed-Integer Linear Programming approach for the automated identification of anomalies in timed discrete systems modeled by Time PNs has been proposed. Using the identification procedure, the nominal model of the system, assumed to be known, can be repaired by adding new transitions to the nominal model and/or extending the firing interval of nominal

TABLE V Number of constraints of $G_I(T_q^{un},\tau_q)$ and $G_F(T_q^{un},T_q^{miss},q)$ for the Case Study, with m=42 and $n_f=3.$

| S | T_q^{un} | T_q^{miss} | Г | $y_{G_I}(\overline{\mathfrak{S}})$ | $y_{G_F}(\overline{\mathfrak{S}})$ |
|-----|------------|--------------|----|------------------------------------|------------------------------------|
| GI | 1 | 0 | 6 | 3 | 1725 |
| S11 | 1 | 0 | 6 | 3 | 1725 |
| S | 1 | 0 | 10 | 3 | 2241 |

transitions. An experimental case study has been presented to show the effectiveness of the approach.

The main drawback of the proposed approach is the computational complexity since the size of the Mixed-Integer Linear Program describing the problem increases with the number of places, with the number of fault transitions and with the length of the observed sequences. Future researches will focus on reducing the complexity when considering particular net structures or using appropriate heuristics, for example looking for suboptimal solutions with respect to the chosen performance index. However, on the basis of some *a priori* knowledge of the system, it has been shown how to reduce the number of unknowns and constraints in order to accelerate significantly the computation time.

On the other hand, the use of the timing information allows to accelerate the repairing process with respect to the untimed approach, thanks to the concept of anomalous firing durations. They occur every time a transition fires before a time less than its firing interval lower bound or after than a time greater than its upper bound is elapsed.

The repairing 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.

Finally, 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 [30] even if they model concurrent activities with no causal relationship.

REFERENCES

- F. Basile, M. P. Cabasino, and C. Seatzu, "State estimation and fault diagnosis of labeled time petri net systems with unobservable transitions," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 997–1009, April 2015.
- [2] 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, 2012.
- [3] C. Ramchandani, "Analysis of asynchronous concurrent systems by timed Petri nets," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1974.
- [4] P. M. Merlin, "A study of the recoverability of computing systems." Ph.D. dissertation, University of California, Irvine, 1974.
- [5] 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, 2012, vol. 99, pp. 169–194.
- [6] W. M. van der Aalst, "Process mining in the large: A tutorial," in *Business Intelligence*, ser. Lecture Notes in Business Information Processing, E. Zimányi, Ed. Springer International Publishing, 2014, vol. 172, pp. 33–76.

- [7] D. Fahland and W. M. van der Aalst, "Repairing process models to reflect reality," in *Business Process Management*, ser. Lecture Notes in Computer Science, A. Barros, A. Gal, and E. Kindler, Eds. Springer Berlin Heidelberg, 2012, vol. 7481, pp. 229–245.
- [8] M. P. 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.
- [9] 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.
- [10] M. P. Cabasino, P. Darondeau, M. P. Fanti, and C. Seatzu, "Model identification and synthesis of discrete-event systems," in *Contemporary Issues in System Science and Engineering*, ser. IEEE/Wiley Press Book Series, 2013.
- [11] 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.
- [12] K. Hiraishi, "Construction of a class of safe Petri nets by presenting firing sequences," in *Lecture Notes in Computer Science*, vol. 616. Springer-Verlag, 1992, pp. 244–262.
- [13] 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.
- [14] P. Darondeau, "Region Based Synthesis of P/T-Nets and Its Potential Applications," in *Lecture Notes in Computer Science*, vol. 1825. Springer-Verlag, 2000, pp. 16–23.
- [15] M. E. Meda-Campana and E. Lopez-Mellado, "Required event sequences for identification of discrete event systems," in 41st Conf. on Decision and Control, Maui, Hawaii, December 2003, pp. 3778–3783.
- [16] 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.
- [17] F. Basile, P. Chiacchio, and J. Coppola, "An approach for the identification of time Petri net systems," in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA'13), Cagliari, Italy*, September 2013, pp. 1–8.
- [18] 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.
- [19] 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.
- [20] M. Dotoli, M. P. Fanti, A. M. Mangini, and W. Ukovich, "Identification of the unobservable behaviour of industrial automation systems by petri nets," *Control Engineering Practice*, vol. 19, no. 9, pp. 958 – 966, 2011.
- [21] A. P. Estrada-Vargas, E. López-Mellado, and J. Lesage, "Input-output identification of controlled discrete manufacturing systems," *Int. J. Systems Science*, vol. 45, no. 3, pp. 456–471, 2014.
- [22] A. P. Estrada-Vargas, E. Lopez-Mellado, and J. J. Lesage, "A blackbox identification method for automated discrete-event systems," to appear on IEEE Transactions on Automation Science and Engineering, doi=10.1109/TASE.2015.2445332, 2015.
- [23] J. Saives, G. Faraut, and J. J. Lesage, "Identification of discrete event systems unobservable behaviour by petri nets using language projections," 2015 European Control Conference (ECC'15), Linz, Austria, pp. 464–471, 2015.
- [24] F. Basile, P. Chiacchio, J. Coppola, and G. De Tommasi, "Identification of Petri nets using timing information," 3rd International Workshop on Dependable Control of Discrete Systems (DCDS'11), Saarbrucken, Germany, pp. 154 –161, 2011.
- [25] F. Basile, P. Chiacchio, and J. Coppola, "Identification of Time Petri net models," *IEEE Transactions on Systems, Man and Cybernetics: Systems,* doi=http://dx.doi.org/10.1109/TSMC.2016.2523929, 2016.
- [26] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.
- [27] C. Seatzu, M. Silva, and J. H. van Schuppen, Eds., *Control of Discrete-Event Systems*, ser. Lecture Notes in Control and Information Sciences. Springer, 2013, vol. 433.
- [28] 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, Oct 2012.
- [29] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, "Task-oriented motion planning for multi-arm robotic systems," *Robotics*

and Computer-Integrated Manufacturing, vol. 28, no. 5, pp. 569 – 582, 2012.

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