Fault Diagnosis of Discrete-Event Systems using Continuous Petri Nets

-draft-

Cristian Mahulea, Carla Seatzu, Maria Paola Cabasino, Manuel Silva [*]

December 23, 2014

Abstract

When discrete event systems are used to model systems with a large number of possible (reachable) states, many problems such as simulation, optimization, and control, may become computationally prohibitive because they require some enumeration of such states. A common way to effectively address this issue is *fluidization*. The goal of this paper is that of studying the effect of fluidization on fault diagnosis. In particular, we focus on the purely logic Petri net model that results in the untimed continuous Petri net model after fluidization. In accordance to most of the literature on discrete event systems, we define three diagnosis states, namely N, U and F, corresponding respectively to *no fault*, *uncertain* and *fault* state. We prove that, given an observation, the resulting diagnosis state can be computed solving linear programming problems rather than integer programming problems as in the discrete case. The main advantage of fluidization is that it enables to deal with much more general Petri net structures. In particular, the unobservable subnet needs not be acyclic as in the discrete case. Moreover, the compact representation of the set of consistent markings using convex polytopes can be seen in some cases as an improvement in terms of computational complexity.

Published as:
C. Mahulea, C. Seatzu, M.P. Cabasino, and M. Silva, "Fault Diagnosis of Discrete-Event Systems using Continuous Petri Nets," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 42, no. 4, pp. 970 - 984, July 2012.. DOI: 10.1109/TSMCA.2012.2183358

[*] C. Mahulea and M. Silva are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain {cmahulea, silva@unizar.es}. M.P. Cabasino and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy {cabasino,seatzu@diee.unica.it}.
This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498). At University of Zaragoza the work has been partially supported also by CICYT - FEDER project DPI2010-20413 and by Fundación Aragón I+D.
This paper is based on our results in [1–3].

1 Introduction

The complexity of nowadays systems makes the problem of deriving efficient approaches for fault diagnosis a major requirement. As a consequence, significant contributions have been proposed in the literature in the last years, dealing with fault detection, isolation and treatment of failures in the case of continuous-time, discrete-time and discrete event systems [4–8]. The idea is to construct fault tolerant models which can detect and adapt their software or hardware in order to allow the system to continue working until repairs can be realistically scheduled.

Faults correspond to discrete events modeling anomalous behaviors. As an example, in a telecommunication system, a fault may correspond to a message that is lost or not sent to the appropriate receiver. In a traffic system, a fault may be a traffic light that does not switch from red to green according to the given schedule. In a manufacturing system it may be the failure of a certain operation, e.g., a wrong assembly, or a part put in a wrong buffer, and so on.

In the literature, faults are often classified as *permanent*, *intermittent* or *control* faults. A fault is "permanent" if its effect remains permanently after its occurrence. On the contrary, "intermittent" faults model faulty behaviors that occur intermittently, with fault events followed by "reset" events, new occurrences of fault events, and so forth [9]. Finally, "control faults" usually model errors of the control system, e.g., software errors. Therefore optimal controllers should be designed so as to tolerate control errors. In practice, the control approaches should be robust so as to avoid violating security specifications also in the presence of fault errors.

Now, according to the above classification, the faults considered in this paper may either be permanent or control faults, while they cannot be intermittent faults since no notion of "reset" is introduced here.

In the case of discrete event systems with a large number of reachable states the problem of fault detection, as well as many others, becomes computationally prohibitive because of the state explosion. A common technique to overcome this is *fluidization*. Several discrete-event based fluid models have been proposed in the literature, some of them derived by the fluidization of queuing networks [10–12] or Petri nets (PNs) [13, 14] [15, 16]. The main idea of the fluidization of PNs is the relaxation of the transitions firings allowing them to fire in positive real amounts. Therefore, the content of the places is no more restricted to take natural values, but it may be expressed by nonnegative real numbers. This implies a series of significant properties. As an example, the reachability set is convex [17]. Moreover, as it will be proved in this paper, in the case of partial observation of the transitions firings (namely in the presence of silent transitions), the set of markings that are consistent with a given observation is convex.

Using this convexity property, the fault detection problem is studied here for untimed continuous Petri nets (CPNs). In particular, in this paper we assume that certain transitions are not observable, including fault transitions and transitions modeling a regular behavior. Thus, faults are only detected on the basis of the observation of a subset of transitions. Fault transitions are partitioned into different fault classes and three different diagnosis states are defined, each one representing a different degree of alarm: N means that *no* fault of a given class has surely occurred; U means that a fault of a given class may have occurred or not (*uncertain* state); F means that a *fault* of a given class has surely occurred. We derive a criterion to define, for each fault class, the value of the diagnosis state, given the observation of a sequence of transitions firings.

Note that uncertain states are common to all discrete event systems diagnosis approaches. This is a natural consequence of partial events observation. Indeed, when the observation of the system behavior is not complete, it may occur that the observed sequence of events is consistent with both a regular and a faulty behavior, thus the resulting diagnosis state is *uncertain*.

In this paper general PN structures are considered and the only assumption made, common to all works dealing with fault diagnosis, is that the unobservable subnet has no spurious markings, i.e., all solutions of the state equation are reachable markings. Since in continuous case this assumption is not very restrictive, this allows one to consider as well unobservable subnets that are cyclic, making the procedure more general with respect to almost all the approaches developed in the discrete event systems framework [18–20].

The paper is organized as follows. In Section 2 a survey on the literature on diagnosis of discrete event systems is presented. Section 3 provides a comparison among the proposed approach and the other approaches mentioned in Section 2. In Section 4 some background on untimed CPNs is given. In Section 5 we introduce the main notations and definitions used in the paper. Then the convexity of a particular set, that is the key point for the proposed diagnosis procedure, is proved and an algorithm to compute it is given. In Section 6 diagnosis states are defined and it is shown how to compute them using linear programming. Two manufacturing examples are considered in Section 7 so as to validate the effectiveness of the procedure. Conclusions are finally drawn in Section 8. In the appendix the main notations used in the paper are reported.

2 Literature review

The diagnosis of discrete event systems is a research area that has received a lot of attention in the last years and has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems. A failure is defined to be any deviation of a system from its normal or intended behavior. Diagnosis is the process of detecting an abnormality in the system behavior and isolating the cause or the source of this abnormality.

In the discrete event systems framework, fault detection has been firstly studied using automata. Interesting contributions have been proposed by Boel and van Schuppen [21], by Debouk *et al.* [22], by Hashtrudi Zad *et al.* [23], by Jiang and Kumar [24], by Lunze and Schroder [25], and by Sampath et al. [26, 27].

More recently this problem has also been addressed in the framework of Petri nets. The intrinsically distributed nature of Petri net models, where the notion of state, i.e., marking, and action, i.e., transition, is local, have often been an asset to reduce the computational complexity involved in solving a diagnosis problem. Among the different contributions in this area we recall the work of Benveniste *et al.* [28], Cabasino *et al.* [19], Dotoli *et al.* [20], Genc and Lafortune [18], Jiroveanu and Boel [29], Lefebvre and Delherm [30] and Ramirez Treviño *et al.* [31].

In particular, Benveniste *et al.* [28] use a net unfolding approach for designing an on-line asynchronous diagnoser. The state explosion is avoided but the on-line computation can be high due to the on-line building of the PN structures by means of the unfolding.

In [19] Cabasino *et al.* present a fault detection approach for discrete event systems using Petri nets, where some transitions of the net are unobservable, including all those transitions that model faulty behaviors. The diagnosis approach is based on the notions of basis marking and justification, that allow one to characterize the set of markings that are consistent with the actual observation, and the set of unobservable transitions whose firing enables it. This approach applies to all net systems whose unobservable subnet is acyclic.

Dotoli *et al* [20] address the on-line fault detection of discrete event systems modeled by Petri nets. The paper recalls a previously proposed diagnoser that works on-line and employs an algorithm based on the definition and solution of some integer linear programming problems to decide whether the system behavior is normal or exhibits some possible faults. To cope with the algorithm computational complexity, they present sufficient conditions guaranteeing that the continuous relaxation of the ILP problems provides an integer solution if the unobservable subnet of the Petri net system considered is an acyclic state machine. In this way the proposed algorithm turns out to exhibit polynomial complexity.

Genc and Lafortune [18] propose a diagnoser on the basis of a modular approach that performs the diagnosis of faults in each module. Subsequently, the diagnosers recover the monolithic diagnosis information obtained when all the modules are combined into a single module that preserves the behavior of the underlying modular system. A communication system connects the different modules and updates the diagnosis information. Even if the approach does not avoid the state explosion problem, an improvement is obtained when the system can be modeled as a collection of PN modules coupled through common places.

Jiroveanu and Boel [29] propose an algorithm for the model based design of a distributed protocol for fault detection and diagnosis of large systems. The overall process is modeled as time PN models that interact with each other via guarded transitions that become enabled only when certain conditions are satisfied. Different local agents receive local observation as well as messages from neighboring agents. Each agent estimates the state of the part of the overall process for which it has model and from which it observes events by reconciling observations with model based predictions. They design algorithms that use limited information exchange between agents and that can quickly decide questions about whether and where a fault occurred and whether or not some components of the local processes have operated correctly. The algorithms they derive allow each local agent to generate a preliminary diagnosis prior to any communication and they show that after the communications among agents the diagnosis performances are the same as in the central case.

Lefebvre and Delherm [30] study the faulty behaviors modeled with ordinary Petri nets with some "fault" transitions. Partial but unbiased measurement of the places marking variation is used in order to estimate the firing sequences. The main contribution is to decide which sets of places must be observed for the exact estimation of some given firing sequences. Minimal diagnosers are defined that detect and isolate the firing of fault transitions immediately.

Ramirez-Treviño *et al.* [31] employ Interpreted PNs to model the system behavior that includes both events and states partially observable. Based on the Interpreted PN model derived from an on-line methodology, a scheme utilizing a solution of a programming problem is proposed to solve the problem of diagnosis.

3 A comparison among the proposed approach and other methods in the literature

Let us now discuss the main differences among the proposed fault diagnosis approach and the ones mentioned in the previous section.

The first main difference consists in the assumed model. In fact, in this paper we consider untimed continuous Petri nets while in [18–20, 28, 30] discrete Petri nets are taken into account; [29] focuses on timed Petri nets and [31] on interpreted Petri nets. Moreover in Ramirez-Treviño *et al.* [31] continuous information on the marking of some places are given, while in [30] the authors deal with ordinary Petri nets, and in [18–20] the assumption on the acyclicity of the unobservable subnet has to be satisfied. Finally, in [18, 29] the authors propose distributed techniques for diagnosis while here we are considering a centralized approach.

The second main difference with respect to all the fault diagnosis approaches presented in the discrete event systems literature, not only based on Petri nets, but on automata as well, is that to the best of our knowledge the proposed procedure is the only one that can also be applied to systems whose unobservable part contains cycles. This obviously consists in a significant advantage in terms of generality of the method.

The other important aspect that should be considered to evaluate the effectiveness of the proposed technique is the computational complexity and in particular the number of information that should be kept into account. Unfortunately, it is not so easy, and probably nonsense, to compare the fluid approach with an arbitrary other one, based on a different model and on different assumptions. What we have done in this paper is to compare the proposed procedure with the approach for discrete nets we presented in [19], based on the notion of basis markings. Note that we believe such a comparison significant since the technique in [19] is known to present significant advantages in terms of computational complexity since it does not require an exhaustive enumeration of the system states, but only a subset of it.

As a result of such a comparison, we conclude that, as it will be shown in Section 7, the computational complexity of both procedures depends on the particular net structure, on the observed word and on the initial marking as well, thus a general claim cannot be given in this respect. Nevertheless, there exist cases in which the proposed method provides a considerable improvement on the computational complexity also allowing to deal with cases that cannot be dealt with the discrete framework.

Summarizing, the conclusion of our investigation is that fluidization is basically suggested in two cases. The first one is when the unobservable subnet is cyclic, being in such a case the only viable approach. The second case is when the advantages in terms of computational complexity are really significant such as in the case of systems with a very large number of reachable states as in the manufacturing example in Subsection 7.2.

4 Background on Untimed CPNs

In this section we provide the basic background on untimed CPNs. For more details we address to [13, 14].

Definition 1 *A CPN system is a pair $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, where:*

- *$\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is the net structure with two disjoint sets of places P and transitions T; pre and post incidence matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}_{\geq 0}^{|P| \times |T|}$, denote the weight of the arcs from places to transitions (respectively, transitions to places);*
- *$\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the initial marking.* ■

Let $q = |P|$ and $n = |T|$ be the cardinality of the set of places and transitions, respectively.

The input and output set of a node $x \in P \cup T$ is denoted ${}^\bullet x$ and $x^\bullet$, respectively. The token load of a place p_i at the marking $\boldsymbol{m}$ is represented as $\boldsymbol{m}(p_i)$ or simply by m_i.

A transition $t_j \in T$ is enabled at a marking $\boldsymbol{m}$ if $\forall p_i \in {}^\bullet t_j$, $\boldsymbol{m}(p_i) \geq 0$ and the enabling degree of t_j at $\boldsymbol{m}$ is:

$$enab(t_j, \boldsymbol{m}) = \min_{p_i \in {}^\bullet t_j} \frac{m_i}{Pre(p_i, t_j)}. \tag{1}$$

When a transition t_j is enabled at a marking $\boldsymbol{m}$ it can be fired. The main difference with respect to discrete PNs is that in the case of CPNs it can be fired in any real amount α, with $0 < \alpha \leq enab(t_j, \boldsymbol{m})$ and it is not limited to a natural number. Such a firing yields to a new marking $\boldsymbol{m'} = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}(\cdot, t_j)$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow matrix* (or *incidence matrix*). This firing is also denoted $\boldsymbol{m}[t_j(\alpha)\rangle \boldsymbol{m'}$.

If a marking $\boldsymbol{m}$ is reachable from the initial marking through a firing sequence

$$\sigma = t_{r1}(\alpha_1) t_{r2}(\alpha_2) \cdots t_{rk}(\alpha_k),$$

and we denote $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ the *firing count vector* whose component associated to a transition t_j is:

$$\sigma_j = \sum_{h \in H(\sigma, t_j)} \alpha_h$$

where

$$H(\sigma, t_j) = \{h = 1, \ldots, k | t_{r_h} = t_j\},$$

then we can write $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, which is called the *fundamental equation* or *state equation.*

The set of all firable sequences is $\mathcal{L}(\mathcal{N}, \boldsymbol{m}_0)$, while the set of all markings that are reachable with a finite firing sequence is $\mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$. An interesting property of $\mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$ is that it is a *convex set* [17]. That is, if two markings $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are reachable, then any marking

$$\boldsymbol{m}_3 = \alpha \cdot \boldsymbol{m}_1 + (1 - \alpha) \cdot \boldsymbol{m}_2,$$

is also reachable $\forall \alpha \in [0, 1]$.

A net system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is *bounded* if there exists a positive constant k such that, for $\boldsymbol{m} \in \mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$, $m(p) \leq k$.

The net $\mathcal{N}$ is called *consistent* iff $\exists\ \boldsymbol{x} > \boldsymbol{0}$ such that $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$, i.e., it is consistent iff there exists at least a complete sequence, i.e., that considers all transitions, whose firing vector $\boldsymbol{x}$ does not lead to a variation in the actual marking.

A CPN $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is a *marked graph* if $\forall p \in P$, $|{}^\bullet p| = |p^\bullet| \leq 1$ and $Pre(p,t), Post(p,t) \in \{0,1\}$ for any $p \in P$ and any $t \in T$.

Dually, a CPN $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is a *state machine* if $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| \leq 1$ and $Pre(p,t), Post(p,t) \in \{0,1\}$ for any $p \in P$ and any $t \in T$.

Given a net $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$, and a subset $T' \subseteq T$ of its transitions, the *$T'-$induced subnet of $\mathcal{N}$* is the new net $\mathcal{N}' = \langle P, T', \boldsymbol{Pre}', \boldsymbol{Post}' \rangle$ where $\boldsymbol{Pre}', \boldsymbol{Post}'$ are the restrictions of $\boldsymbol{Pre}, \boldsymbol{Post}$ to T'. The net $\mathcal{N}'$ can be thought as obtained from $\mathcal{N}$ removing all transitions in $T \setminus T'$ (and isolated place).

Let T^* be the set of all possible sequences obtainable combining elements in T, included the empty word[1]. Given a subset $T' \subseteq T$, the projection Π of a sequence $\sigma \in T^*$ over T' is defined as $\Pi : T^* \rightarrow T'^*$ such that:

(i) $\Pi(\varepsilon) = \varepsilon$, where ε denotes the empty word;

(ii) for all $\sigma \in T^*$ and $t \in T$, $\Pi(\sigma t) = \Pi(\sigma)t$ if $t \in T'$, and $\Pi(\sigma t) = \Pi(\sigma)$ otherwise.

Given a sequence $\sigma \in \mathcal{L}(\mathcal{N}, \boldsymbol{m}_0)$, we denote $w = \Pi_o(\sigma)$ the corresponding *observed word*, i.e., the projection of σ over the set of *observable* transitions T_o.

In the following, with a little abuse of notation, we will write that $w \in T_o^*$, where T_o is the set of observable transitions as specified in the following section.

Analogously, we denote $\Pi_u(\sigma)$ the unobservable projection of σ, namely its projection over the set of *unobservable* transitions $T_u = T \setminus T_o$.

Let $\boldsymbol{C}_o$ ($\boldsymbol{C}_u$) be the restriction of the incidence matrix to T_o (T_u), namely the matrix obtained from the incidence matrix $\boldsymbol{C}$ removing all columns not relative to transitions in T_o (T_u).

Finally, in the following the T_u-induced subnet will also be called the *unobservable subnet.*

5 The set of $\boldsymbol{y}$-vectors

Let us introduce the notion of $\boldsymbol{y}$-vectors on which our diagnosis approach is based on. We consider the following basic assumptions.

(A1) The initial marking of the net system is known.

(A2) The set of transitions is partitioned as $T = T_o \cup T_u$.

(A3) The T_u-induced net has no *spurious solutions.*

A *spurious marking* is a non reachable marking solution of the state equation, i.e., there exists no firing sequence corresponding to the firing vector. The following proposition provides constructive criteria to establish the validity of Assumption (A3).

[1] The notation T^* is used here with a little abuse. Indeed in the continuous case, to each transition firing is associated a firing amount. Thus T^* denotes the possible sequences obtainable combining elements in T, where each sequence is characterized by the firing amounts of all the transitions in it.

Proposition 2 *Let $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ be a CPN system. All markings $\boldsymbol{m}\in\mathbb{R}^m_{\geq 0}:\boldsymbol{m}=\boldsymbol{m}_0+C\cdot\boldsymbol{\sigma}$, with $\boldsymbol{\sigma}\geq\boldsymbol{0}$, are reachable, i.e., $\mathcal{N}$ has no spurious solution, if at least one of the following three conditions is satisfied:*

- $\mathcal{N}$ *is* acyclic*;*
- $\mathcal{N}$ *is* consistent *and all transitions can fire at* $\boldsymbol{m}_0$*;*
- $\boldsymbol{m}>\boldsymbol{0}$.

Proof: The first item can be proved following exactly the same arguments of Theorem 16 in [32] where the result is proved for discrete Petri nets, with the only difference that in the continuous case the restriction to natural numbers of the firing amount is relaxed.

The second item has been proved in Theorem 3 of [17]. Finally, the third item has been proved in the first item of Corollary 18 in [33]. □

The third assumption, characteristic for continuous nets, states that the interior points of the polytope of the markings solution of the state equation are reachable markings. This condition allows one to deal with a larger class of Petri nets with respect to the discrete case. In particular, nets having the *unobservable subnet cyclic* can be studied (see Subsection 7.1 as an example).

Definition 3 Let $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ be a CPN system where $\mathcal{N}=\langle P,T,\boldsymbol{Pre},\boldsymbol{Post}\rangle$ and $T=T_o\cup T_u$. Let $w\in T_o^*$ be an observed word. We define the *set of firing sequences consistent with* w by

$$\mathcal{L}(w) \quad = \quad \{\sigma\in\mathcal{L}(\mathcal{N},\boldsymbol{m}_0)\ \mid\ \Pi_o(\sigma)=w\} \tag{2}$$

and the *set of markings consistent with* w by

$$\mathcal{C}(w)=\{\boldsymbol{m}\in\mathbb{R}^q_{\geq 0}\ \mid\ \exists\sigma\in T^*:\boldsymbol{m}_0[\sigma\rangle\boldsymbol{m},\ \Pi_o(\sigma)=w\}. \tag{3}$$

■

Definition 4 Let $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ be a CPN system where $\mathcal{N}=\langle P,T,\boldsymbol{Pre},\boldsymbol{Post}\rangle$ and $T=T_o\cup T_u$. Let $\sigma\in\mathcal{L}(\mathcal{N},\boldsymbol{m}_0)$ be a firable sequence and $w=\Pi_o(\sigma)$ the corresponding observed word.

The set of *unobservable sequences consistent with* w is:

$$\Gamma(w)=\{\sigma_u\in T_u^*\mid\quad\exists\sigma\in\mathcal{L}(w)\ :\sigma_u=\Pi_u(\sigma)\}\,. \tag{4}$$

The corresponding *set of* $\boldsymbol{y}$*-vectors* is:

$$\begin{aligned}\overline{Y}(\boldsymbol{m}_0,w)=\{[\boldsymbol{m}^T;\ \boldsymbol{\varrho}^T]^T\mid\ \exists\sigma\in\mathcal{L}(w),\ \Pi_u(\sigma)\in\Gamma(w),\\ \varrho=\Pi_u(\sigma),\ \boldsymbol{m}=\boldsymbol{m}_0+\boldsymbol{C}\cdot\boldsymbol{\sigma}\},\end{aligned} \tag{5}$$

while the set of ϱ-vectors is

$$Y(\boldsymbol{m}_0,w)=\left\{\boldsymbol{\varrho}\in\mathbb{R}^{n_u}\ \mid\ \left[\begin{array}{c}\boldsymbol{m}\\ \boldsymbol{\varrho}\end{array}\right]\in\overline{Y}(\boldsymbol{m}_0,w)\right\}. \tag{6}$$

■

In simple words,

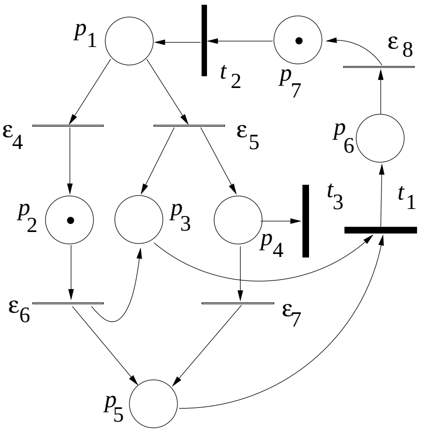

Figure 1: The Petri net system considered in Examples 5, 8, 10 and 12.

- $\Gamma(w)$ is the set of sequences of unobservable transitions interleaved with w whose firing enables w.
- $\overline{Y}(\boldsymbol{m}_0, w)$ is the set of $\boldsymbol{y}$-vectors, where:
  - the first $q = |P|$ entries of the generic vector $\boldsymbol{y}$ coincide with a *consistent marking* $\boldsymbol{m}$, i.e., a possible marking of the system after the observation w;
  - the last $n_u = |T_u|$ entries correspond to the firing count vector of the unobservable sequence that has fired, interleaved with w, in order to reach the consistent marking $\boldsymbol{m}$ from $\boldsymbol{m}_0$. They define a set of vectors that will be called *ϱ-vectors* in the rest of the paper.

Example 5 Let us consider the CPN system in Fig. 1 where

$$T_o = \{t_1, t_2, t_3\}, \quad T_u = \{\varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7, \varepsilon_8\}.$$

Let us first assume that no transition is observed, thus $w = \varepsilon$. In such a case $\Gamma(w) = \{\varepsilon_6(\alpha)\}$, where $\alpha \in [0, 1]$. In fact, ε_6 is the only unobservable transition enabled at $\boldsymbol{m}_0$ and it can fire for any amount $\alpha \in [0, 1]$.

Therefore both

$$\boldsymbol{y}_1 = [0\ 1\ 0\ 0\ 0\ 0\ 1 \mid 0\ 0\ 0\ 0\ 0]^T,$$

and

$$\boldsymbol{y}_2 = [0\ 0\ 1\ 0\ 1\ 0\ 1 \mid 0\ 0\ 1\ 0\ 0]^T$$

belong to the set $\overline{Y}(\boldsymbol{m}_0, \varepsilon)$. In particular, $\boldsymbol{y}_1$ corresponds to $\alpha = 0$, while $\boldsymbol{y}_2$ corresponds to $\alpha = 1$. Indeed the first q entries of $\boldsymbol{y}_1$ coincide with the initial marking, while its last n_u entries are null. Finally, the first q components of $\boldsymbol{y}_2$ coincide with the marking reached firing $\varepsilon_6(1)$, while its last n_u entries correspond to the firing count vector of the unobservable transitions: $[\varrho(\varepsilon_4)\ \varrho(\varepsilon_5)\ \varrho(\varepsilon_6)\ \varrho(\varepsilon_7)\ \varrho(\varepsilon_8)]^T = [0\ 0\ 1\ 0\ 0]^T$.

As it will be formally proved in Proposition 6, all vectors $\boldsymbol{y}$ obtained as a convex combination of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are $\boldsymbol{y}$-vectors as well.

Now, let us assume that $t_1(0.7)$ is observed, i.e., $w = t_1(0.7)$. For sure ε_6 has fired at least for an amount $\alpha = 0.7$ before w since its firing is the only way to enable $t_1(0.7)$. However, after the firing of t_1, transition ε_8 may fire for an amount $\alpha' \in [0, 0.7]$ while ε_6 can be fired in any amount $\alpha'' \in [0, 0.3]$. Hence, $\Gamma(t_1(0.7)) = \{\varepsilon_6(0.7 + \alpha''), \varepsilon_6(0.7 + \alpha'')\varepsilon_8(\alpha'), \varepsilon_6(0.7)\varepsilon_8(\alpha')\varepsilon_6(\alpha''), \ldots\}$ where $\alpha' \in [0, 0.7]$, $\alpha'' \in [0, 0.3]$ and dots denote all other sequences of unobservable transitions with the same firing vector as the previous ones. Repeating the same arguments as in the previous case, we can conclude that the following four vectors all belong to $\overline{Y}(\boldsymbol{m}_0, t_1(0.7))$

$$\begin{aligned}
\boldsymbol{y}'_1 &= [0\ 0\ 0.3\ 0\ 0.3\ 0\ 1.7 \mid 0\ 0\ 1\ 0\ 0.7]^T,\\
\boldsymbol{y}'_2 &= [0\ 0.3\ 0\ 0\ 0\ 0\ 1.7 \mid 0\ 0\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}'_3 &= [0\ 0.3\ 0\ 0\ 0\ 0.7\ 1 \mid 0\ 0\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}'_4 &= [0\ 0\ 0.3\ 0\ 0.3\ 0.7\ 1 \mid 0\ 0\ 1\ 0\ 0]^T.
\end{aligned}$$

■

Proposition 6 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a CPN system where $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ and $T = T_o \cup T_u$.*

Given an observable transition $t \in T_o$ firing an amount α, under assumption (A3), the set $\overline{Y}(\boldsymbol{m}_0, w)$ is convex.

Proof: Let us rewrite the observed sequence as

$$w = t_{r_1}(\alpha_1) t_{r_2}(\alpha_2) \ldots t_{r_k}(\alpha_k). \tag{7}$$

Moreover, let

$$\sigma' = \sigma'_{u_1} t_{r_1}(\alpha_1) \sigma'_{u_2} t_{r_2}(\alpha_2) \ldots \sigma'_{u_k} t_{r_k}(\alpha_k) \sigma'_{u_{k+1}}$$

and

$$\sigma'' = \sigma''_{u_1} t_{r_1}(\alpha_1) \sigma''_{u_2} t_{r_2}(\alpha_2) \ldots \sigma''_{u_k} t_{r_k}(\alpha_k) \sigma''_{u_{k+1}}$$

be two sequences whose observable projections are equal to w, being

$$\sigma'_{u_1}, \sigma''_{u_1}, \ldots, \sigma'_{u_k}, \sigma''_{u_k}, \sigma'_{u_{k+1}}, \sigma''_{u_{k+1}} \in T_u^*.$$

Assume that σ' and σ'' are both enabled at $\boldsymbol{m}_0$. Thus, by definition,

$$\boldsymbol{y}' = \begin{bmatrix} \boldsymbol{m}' \\ \boldsymbol{\varrho}' \end{bmatrix} \in \overline{Y}(\boldsymbol{m}_0, w)$$

if

$$\begin{cases} \boldsymbol{m}' = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}', \\ \boldsymbol{\varrho}' = \boldsymbol{\sigma}'_{u_1} + \boldsymbol{\sigma}'_{u_2} + \ldots + \boldsymbol{\sigma}'_{u_k} + \boldsymbol{\sigma}'_{u_{k+1}}, \end{cases}$$

and

$$\boldsymbol{y}'' = \begin{bmatrix} \boldsymbol{m}'' \\ \boldsymbol{\varrho}'' \end{bmatrix} \in \overline{Y}(\boldsymbol{m}_0, w)$$

if

$$\begin{cases} \boldsymbol{m}'' = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}'', \\ \boldsymbol{\varrho}'' = \boldsymbol{\sigma}''_{u_1} + \boldsymbol{\sigma}''_{u_2} + \ldots + \boldsymbol{\sigma}''_{u_k} + \boldsymbol{\sigma}''_{u_{k+1}}. \end{cases}$$

We want to prove that any convex combination of $\boldsymbol{y}'$ and $\boldsymbol{y}''$ still belongs to $\overline{Y}(\boldsymbol{m}_0, w)$.

To this aim let $\delta, \beta \in [0,1]$ such that $\delta + \beta = 1$. Being the net system continuous, by assumption (A3) it holds

$$\begin{array}{rl} & \boldsymbol{m}_0 + \delta \cdot \boldsymbol{C}_u \cdot \boldsymbol{\sigma}'_{u_1} + \beta \cdot \boldsymbol{C}_u \cdot \boldsymbol{\sigma}''_{u_1} \\ = & \delta(\boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}'_{u_1}) + \beta(\boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}''_{u_1}) \\ \geq & \delta \cdot \alpha_1 \cdot \boldsymbol{Pre}(\cdot, t_{r_1}) + \beta \cdot \alpha_1 \cdot \boldsymbol{Pre}(\cdot, t_{r_1}) \\ = & \alpha_1 \cdot \boldsymbol{Pre}(\cdot, t_{r_1}), \end{array}$$

thus

$$\boldsymbol{y}_1 = \left[\begin{array}{c} \boldsymbol{m}_1 \\ \boldsymbol{\varrho}_1 \end{array} \right] \in \overline{Y}(\boldsymbol{m}_0, t_{r_1}(\alpha_1))$$

if

$$\left\{ \begin{array}{l} \boldsymbol{m}_1 = \boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\varrho}_1 + \alpha_1 \cdot \boldsymbol{C}(\cdot, t_{r_1}), \\ \boldsymbol{\varrho}_1 = \delta \boldsymbol{\sigma}'_{u_1} + \beta \boldsymbol{\sigma}''_{u_1}. \end{array} \right.$$

Analogously,

$$\begin{array}{rl} & \boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\varrho}_1 + \boldsymbol{C}_o \cdot \boldsymbol{\sigma}_{r_1} + \delta \cdot \boldsymbol{C}_u \cdot \boldsymbol{\sigma}'_{u_2} + \beta \cdot \boldsymbol{C}_u \cdot \boldsymbol{\sigma}''_{u_2} \\ = & \delta(\boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\varrho}_1 + \boldsymbol{C}_o \cdot \boldsymbol{\sigma}_{r_1} + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}'_{u_2}) + \\ & \beta(\boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\varrho}_1 + \boldsymbol{C}_o \cdot \boldsymbol{\sigma}_{r_1} + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}''_{u_2}) \\ \geq & \delta \cdot \alpha_2 \cdot \boldsymbol{Pre}(\cdot, t_{r_2}) + \beta \cdot \alpha_2 \cdot \boldsymbol{Pre}(\cdot, t_{r_2}) \\ = & \alpha_2 \cdot \boldsymbol{Pre}(\cdot, t_{r_2}), \end{array}$$

thus

$$\boldsymbol{y}_2 = \left[\begin{array}{c} \boldsymbol{m}_2 \\ \boldsymbol{\varrho}_2 \end{array} \right] \in \overline{Y}(\boldsymbol{m}_0, t_{r_1}(\alpha_1) t_{r_2}(\alpha_2))$$

if

$$\left\{ \begin{array}{l} \boldsymbol{m}_2 = \boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\varrho}_2 + \alpha_1 \cdot \boldsymbol{C}(\cdot, t_{r_1}) + \alpha_2 \cdot \boldsymbol{C}(\cdot, t_{r_2}), \\ \boldsymbol{\varrho}_2 = \delta(\boldsymbol{\sigma}'_{u_1} + \boldsymbol{\sigma}'_{u_2}) + \beta(\boldsymbol{\sigma}''_{u_1} + \boldsymbol{\sigma}''_{u_2}). \end{array} \right.$$

Generalizing to a word w of arbitrary length $k \geq 1$ defined as in equation (7), we can conclude that

$$\boldsymbol{y} = \alpha \boldsymbol{y}' + \beta \boldsymbol{y}'' \in \overline{Y}(\boldsymbol{m}_0, w)$$

thus proving the statement. □

If the net system is *bounded* the set $\overline{Y}(\boldsymbol{m}_0, w)$ can be easily characterized in linear algebraic terms. Moreover, if the net system is bounded, even if there exist cycles of unobservable transitions, the enabling degree of the unobservable transitions is upper bounded. In more detail, the *structural enabling bound* of a given transition t of $\mathcal{N}$ is the solution of the following LPP (see [34] for more details):

$$\begin{array}{rll} EN(t) = & \max k & \\ & \text{s.t.} & \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \geq k \cdot \boldsymbol{Pre}(\cdot, t) \\ & & \boldsymbol{\sigma} \geq \boldsymbol{0}. \end{array} \tag{8}$$

Now, let $\boldsymbol{EN} \in \mathbb{R}_{\geq 0}^{|T_u|}$ be a vector with as many entries as the number of unobservable transitions, where each entry is equal to the structural enabling bound of the corresponding unobservable transition. The following algorithm can be used for the characterization of $\overline{Y}(\boldsymbol{m}_0, w)$.

Algorithm 7 (Computation of $\overline{\boldsymbol{Y}}(\boldsymbol{m}_0, \boldsymbol{w})$)

1. Let $v = \varepsilon$.

2. Let $\overline{Y}(\boldsymbol{m}_0, v)$ be the polytope[2] defined as

$$\begin{cases} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}_u \\ \boldsymbol{m} \geq \boldsymbol{0} \\ \boldsymbol{0} \leq \boldsymbol{\sigma}_u \leq \boldsymbol{EN}. \end{cases}$$

3. Let $t(\alpha)$ be a new observation and $w = vt(\alpha)$.

4. Compute the set of vertices $\mathcal{E}(v)$ of

$$\begin{cases} [\boldsymbol{m}^T;\ \boldsymbol{\varrho}^T]^T \in \overline{Y}(\boldsymbol{m}_0, v) \\ \boldsymbol{m} \geq \alpha \cdot \boldsymbol{Pre}(\cdot, t). \end{cases}$$

5. Let $E = \emptyset$.

6. For all $\boldsymbol{e}_i = [\tilde{\boldsymbol{m}}^T;\ \tilde{\boldsymbol{\varrho}}^T]^T \in \mathcal{E}(v)$:

   (a) compute the set of vertices $E_i = [\boldsymbol{m}^T;\ \boldsymbol{\varrho}^T]^T$ of the polytope defined as

   $$\begin{cases} \boldsymbol{m} = \tilde{\boldsymbol{m}} + \alpha \cdot \boldsymbol{C}(\cdot, t) + \boldsymbol{C}_u \cdot \boldsymbol{\sigma}_u \\ \boldsymbol{\varrho} = \tilde{\boldsymbol{\varrho}} + \boldsymbol{\sigma}_u \\ \boldsymbol{0} \leq \boldsymbol{\sigma}_u \leq \boldsymbol{EN} \\ \boldsymbol{m} \geq \boldsymbol{0} \end{cases} \tag{9}$$

   (b) let $E = E \cup E_i$.

7. Let $\overline{Y}(\boldsymbol{m}_0, w)$ be the convex hull of E.

8. Let $Y(\boldsymbol{m}_0, w) = \left\{ \boldsymbol{\varrho} \in \mathbb{R}^{n_u} \mid \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\varrho} \end{bmatrix} \in \overline{Y}(\boldsymbol{m}_0, w) \right\}$.

9. Let $v = w$ and goto Step 3.

■

In simple words Algorithm 7 first computes in Step 2 the set $\overline{Y}(\boldsymbol{m}_0, \varepsilon)$. By definition it includes all firing vectors corresponding to sequences of unobservable transitions that are enabled at the initial marking.

Then, after a new observation $t(\alpha)$ occurs, it computes the set of vertices of $\overline{Y}(\boldsymbol{m}_0, v)$ from which $t(\alpha)$ is enabled, denoted $\mathcal{E}(v)$ where $v = \varepsilon$. Now, for each vertex $\boldsymbol{e}_i = [\tilde{\boldsymbol{m}}^T;\ \tilde{\boldsymbol{\varrho}}^T]^T \in \mathcal{E}(v)$, it defines the set of markings – $\boldsymbol{\varrho}$-vectors that can be obtained from $\tilde{\boldsymbol{m}}$ firing $t(\alpha)$ plus eventually a sequence of unobservable transitions (σ_u). Note that by Assumption (A3) this does not lead to spurious solutions. Then the algorithm computes the set of vertices E_i of such a set. Finally, $\overline{Y}(\boldsymbol{m}_0, t(\alpha))$ is the convex hull of the union of all the vertices thus obtained. The algorithm iterates when a new observation occurs.

Example 8 Let us consider again the CPN system in Fig. 1. Assume that an observation $w = t_1(0.7)t_2(0.5)t_3(0.5)$ occurs. We apply Algorithm 7 to compute the set of vertices of $\overline{Y}(\boldsymbol{m}_0, w)$.

[2] A *bounded polyhedron* $\mathcal{P} \subset \mathbb{R}^n$, $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{B}\}$ is called a *polytope*.

In accordance with the results in Example 5, we obtain that $\overline{Y}(\boldsymbol{m}_0, \varepsilon)$ has two vertices: $\boldsymbol{y}_1 = [0\ 1\ 0\ 0\ 0\ 0\ 1 \mid 0\ 0\ 0\ 0\ 0]^T$ and $\boldsymbol{y}_2 = [0\ 0\ 1\ 0\ 1\ 0\ 1 \mid 0\ 0\ 1\ 0\ 0]^T$.

Using Algorithm 7 we also compute the set of vertices of $\overline{Y}(\boldsymbol{m}_0, t_1(0.7))$:

$$\begin{aligned}
\boldsymbol{y}'_1 &= [0\ 0\ 0.3\ 0\ 0.3\ 0\ 1.7 \mid 0\ 0\ 1\ 0\ 0.7]^T,\\
\boldsymbol{y}'_2 &= [0\ 0.3\ 0\ 0\ 0\ 0\ 1.7 \mid 0\ 0\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}'_3 &= [0\ 0.3\ 0\ 0\ 0\ 0.7\ 1 \mid 0\ 0\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}'_4 &= [0\ 0\ 0.3\ 0\ 0.3\ 0.7\ 1 \mid 0\ 0\ 1\ 0\ 0]^T.
\end{aligned}$$

Iterating the procedure we find out a set of 16 vertices defining $\overline{Y}(\boldsymbol{m}_0, t_1(0.7)t_2(0.5))$, namely

$$\begin{aligned}
\boldsymbol{y}''_1 &= [0\ 0.3\ 0.5\ 0\ 0.5\ 0\ 1.2 &&\mid 0\ 0.5\ 0.7\ 0.5\ 0.7]^T,\\
\boldsymbol{y}''_2 &= [0.5\ 0.3\ 0\ 0\ 0\ 0\ 1.2 &&\mid 0\ 0\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}''_3 &= [0\ 0.3\ 0.5\ 0.5\ 0\ 0\ 1.2 &&\mid 0\ 0.5\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}''_4 &= [0\ 0.8\ 0\ 0\ 0\ 0\ 1.2 &&\mid 0.5\ 0\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}''_5 &= [0\ 0\ 0.8\ 0\ 0.8\ 0\ 1.2 &&\mid 0.5\ 0\ 1.5\ 0\ 0.7]^T,\\
\boldsymbol{y}''_6 &= [0\ 0\ 0.8\ 0.5\ 0.3\ 0\ 1.2 &&\mid 0\ 0.5\ 1\ 0\ 0.7]^T,\\
\boldsymbol{y}''_7 &= [0.5\ 0\ 0.3\ 0\ 0.3\ 0\ 1.2 &&\mid 0\ 0\ 1\ 0\ 0.7]^T,\\
\boldsymbol{y}''_8 &= [0\ 0\ 0.8\ 0\ 0.8\ 0\ 1.2 &&\mid 0\ 0.5\ 1\ 0.5\ 0.7]^T,\\
\boldsymbol{y}''_9 &= [0\ 0\ 0.8\ 0\ 0.8\ 0.7\ 0.5 &&\mid 0.5\ 0\ 1.5\ 0\ 0]^T,\\
\boldsymbol{y}''_{10} &= [0\ 0\ 0.8\ 0.5\ 0.3\ 0.7\ 0.5 &&\mid 0\ 0.5\ 1\ 0\ 0]^T,\\
\boldsymbol{y}''_{11} &= [0.5\ 0\ 0.3\ 0\ 0.3\ 0.7\ 0.5 &&\mid 0\ 0\ 1\ 0\ 0]^T,\\
\boldsymbol{y}''_{12} &= [0\ 0\ 0.8\ 0\ 0.8\ 0.7\ 0.5 &&\mid 0\ 0.5\ 1\ 0.5\ 0]^T,\\
\boldsymbol{y}''_{13} &= [0\ 0.8\ 0\ 0\ 0\ 0.7\ 0.5 &&\mid 0.5\ 0\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}''_{14} &= [0\ 0.3\ 0.5\ 0.5\ 0\ 0.7\ 0.5 &&\mid 0\ 0.5\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}''_{15} &= [0.5\ 0.3\ 0\ 0\ 0\ 0.7\ 0.5 &&\mid 0\ 0\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}''_{16} &= [0\ 0.3\ 0.5\ 0\ 0.5\ 0.7\ 0.5 &&\mid 0\ 0.5\ 0.7\ 0.5\ 0]^T.
\end{aligned}$$

Note that there are two vertices relative to the same consistent marking, namely $\boldsymbol{y}''_9$ and $\boldsymbol{y}''_{12}$. The reason of this is that the same marking can be obtained by firing two unobservable sequences having different firing vectors. More precisely, $\boldsymbol{m} = [0\ 0\ 0.8\ 0\ 0.8\ 0.7\ 0.5]^T$ can be obtained from $\boldsymbol{m}_0$ firing $\sigma_1 = \varepsilon_6(1)t_1(0.7)t_2(0.5)\varepsilon_4(0.5)\varepsilon_6(0.5)$ or $\sigma_2 = \varepsilon_6(1)t_1(0.7)t_2(0.5)\varepsilon_5(0.5)\ \varepsilon_7(0.5)$.

Finally, after the observation of $t_3(0.5)$, the set of vertices of $\overline{Y}(\boldsymbol{m}_0, t_1(0.7)t_2(0.5)t_3(0.5))$ is reduced to four, namely

$$\begin{aligned}
\boldsymbol{y}_1''' &= [0\ 0\ 0.8\ 0\ 0.3\ 0\ 1.2 &&\mid 0\ 0.5\ 1\ 0\ 0.7]^T,\\
\boldsymbol{y}_2''' &= [0\ 0.3\ 0.5\ 0\ 0\ 0\ 1.2 &&\mid 0\ 0.5\ 0.7\ 0\ 0.7]^T,\\
\boldsymbol{y}_3''' &= [0\ 0.3\ 0.5\ 0\ 0\ 0.7\ 0.5 &&\mid 0\ 0.5\ 0.7\ 0\ 0]^T,\\
\boldsymbol{y}_4''' &= [0\ 0\ 0.8\ 0\ 0.3\ 0.7\ 0.5 &&\mid 0\ 0.5\ 1\ 0\ 0]^T.
\end{aligned}$$

We can conclude that $\varepsilon_6(0.7)$ must have fired before the observation of $t_1(0.7)$ and $\varepsilon_5(0.5)$ must have fired before $t_3(0.5)$. ■

By looking at this very simple example, we can conclude that the number of vertices of $\overline{Y}(\boldsymbol{m}_0, w)$ can either increase or decrease. However, it keeps bounded if the net system is bounded.

6 Fault diagnoser design

Assume that a certain number of *anomalous* (or *fault*) behaviors may occur in the system. The occurrence of a fault behavior corresponds to the firing of an unobservable transition, but there may also be other transitions that are unobservable as well, but whose firing corresponds to regular behaviors. Then, assume that fault behaviors may be divided into r main classes (*fault classes*), and we are not interested in distinguishing among fault events in the same class. Usually, fault transitions that belong to the same fault class are transitions that represent similar physical faulty behavior.

This can be easily modeled in PN terms assuming that the set of unobservable transitions is partitioned into two subsets, namely

$$T_u = T_f \cup T_{reg}$$

where T_f includes all fault transitions and T_{reg} includes all transitions relative to unobservable but regular events. The set T_f is further partitioned into r subsets, namely,

$$T_f = T_f^1 \cup T_f^2 \cup \ldots \cup T_f^r$$

where all transitions in the same subset correspond to the same fault class. We will say that the i-th fault has occurred when a transition in T_f^i has fired.

Let us now introduce the definition of diagnoser.

Definition 9 A *diagnoser* is a function

$$\Delta : T_o^* \times \{T_f^1, T_f^2, \ldots, T_f^r\} \rightarrow \{N, U, F\}$$

that associates to each observation w and to each fault class T_f^i, $i = 1, \ldots, r$, a *diagnosis state*.

- $\Delta(w, T_f^i) = N$ if for all $\sigma \in \mathcal{L}(w)$ and for all $t_f \in T_f^i$ it holds $t_f \notin \sigma$.

  In such a case the ith fault cannot have occurred, because *none* of the firing sequences consistent with the observation contains fault transitions of class i.

- $\Delta(w, T_f^i) = U$ if:
  (i) there exists $\sigma \in \mathcal{L}(w)$ and $t_f \in T_f^i$ such that $t_f \in \sigma$ but
  (ii) there exists $\sigma' \in \mathcal{L}(w)$ such that for all $t_f \in T_f^i$ it holds $t_f \notin \sigma'$

  In such a case a fault transition of class i may have occurred or not, i.e., it is uncertain, and we have no criteria to draw a conclusion in this respect.

- $\Delta(w, T_f^i) = F$ if for all $\sigma \in \mathcal{L}(w)$ there exists $t_f \in T_f^i$ such that $t_f \in \sigma$.

  In such a case the ith fault must have occurred, because all firable sequences consistent with the observation contain at least one fault transition of class i. ■

Thus, states N and F correspond to "certain" states: the fault has not occurred or it has occurred for sure; on the contrary state U is an "uncertain" state: the fault may either have occurred or not.

Example 10 Let us consider again the CPN system in Fig. 1. Assume that there exists only one fault class: $T_f^1 = \{\varepsilon_5\}$.

Obviously, before any observation, $\Delta(\varepsilon, T_f^1) = N$ since there exists no sequence enabled at the initial marking including no observable transition and the fault ε_5.

Now, let $w = t_1(0.7)t_2(0.5)$. The vertices of the set $\overline{Y}(\boldsymbol{m}_0, w)$ are given in Example 8. It is easy to observe that ε_5 may have fired in an amount of 0.5 (see $\boldsymbol{y}''_1$, $\boldsymbol{y}''_3$, $\boldsymbol{y}''_6$, $\boldsymbol{y}''_8$, $\boldsymbol{y}''_{10}$, $\boldsymbol{y}''_{12}$, $\boldsymbol{y}''_{14}$ and $\boldsymbol{y}''_{16}$) or not (e.g. $\boldsymbol{y}''_2$). This implies that $\Delta(t_1(0.7)t_2(0.5), T_f^1) = U$, i.e., the fault may have occurred or not. ■

The on-line computation of the sets $\mathcal{L}(w)$ and $\Gamma(w)$ may be computationally demanding in large scale systems. In the following we suggest an alternative procedure to compute diagnosis states that is based on the knowledge of the set of ϱ-vectors $Y(\boldsymbol{m}_0, w)$.

Proposition 11 *Consider an observed word $w \in T_o^*$. Let*

$$\begin{cases} l_i = \min \sum\limits_{t_j \in T_f^i} \varrho(t_j) \\ s.t. \\ \boldsymbol{\varrho} \in Y(\boldsymbol{m}_0, w) \end{cases} \qquad \begin{cases} u_i = \max \sum\limits_{t_j \in T_f^i} \varrho(t_j) \\ s.t. \\ \boldsymbol{\varrho} \in Y(\boldsymbol{m}_0, w) \end{cases} \tag{10}$$

It holds:

$$\begin{array}{lcl} \Delta(w, T_f^i) = N & \Leftrightarrow & u_i = 0 \\ \Delta(w, T_f^i) = U & \Leftrightarrow & l_i = 0 \wedge u_i > 0 \\ \Delta(w, T_f^i) = F & \Leftrightarrow & l_i > 0 \end{array}$$

Proof: It follows from Definitions 4 and 9.

If $u_i = 0$ it means that none of the unobservable sequences consistent with w contains transitions in T_f^i. By Definition 9 this corresponds to diagnosis state N. Moreover, if $u_i > 0$ it means that at least one unobservable sequence consistent with w contains at least one transition in the ith class, thus the diagnosis state cannot be N.

If $l_i = 0$ and $u_i > 0$ it means that there exist at least one sequence of unobservable transitions consistent with w that does not contain transitions in the ith class and at least one sequence of unobservable transitions consistent with w that contains transitions in the ith class. By definition this is the case of diagnosis state equal to U. Similarly, if any of such conditions is violated the diagnosis state cannot be equal to U.

Finally, if $l_i > 0$ then all the unobservable sequences consistent with w contain at least one transition in T_f^i, i.e., all words consistent with the actual observation contain a transition in the ith class, that means that some fault in the ith class has occurred for sure. By Definition 9 this corresponds to diagnosis state equal to F. Similarly, if $l_i = 0$ it means that some unobservable sequences consistent with w contain no transition in T_f^i thus the diagnosis state is either N or U.

□

Example 12 Let us still consider the CPN in Fig. 1. Assume again that there is only one fault class $T_f^1 = \{\varepsilon_5\}$.

Solving the LPPs (10) it is immediate to obtain the following diagnosis states:

$$\begin{array}{rcl} \Delta(\varepsilon, T_f^1) & = & N \\ \Delta(t_1(0.7), T_f^1) & = & N \\ \Delta(t_1(0.7)t_2(0.5), T_f^1) & = & U \\ \Delta(t_1(0.7)t_2(0.5)t_3(0.5), T_f^1) & = & F. \end{array}$$

■

Note that the numerical results in Examples 5, 8, 10 and 12 have been obtained using the software in [35].

6.1 Some remarks related to fluidization

It is well known that the set of reachable markings of the discrete net is included in the set of reachable markings of the underling continuous one. However, there may exist integer markings in the reachability set of the continuous net that are not reachable in the discrete one. The same result can be easily proved for the set of markings consistent with a given observation. This implies that even if a fault has occurred in the original net and it would have been detected using the discrete approach, it may happen that using the continuous approach we do not detect it, and the output is an uncertain state. On the contrary, if a fault is detected in the continuous case, then for sure it has occurred in the original net.

Obviously, this is a drawback of fluidization. However, in many cases, fluidization is the only viable solution, either because the unobservable subnet is cyclic, or because the computational complexity of the discrete approach is prohibitive as discussed in the following Section 7 via a numerical example. In simple words, it is the same kind of limitation we met when using linear programming to solve integer programming problems.

However there exist some cases in which the above limitation does not appear. In particular, we can prove that under particular assumptions on the net structure, e.g. *total unimodularity* of the incidence matrix, the diagnosis states in the two cases are guaranteed to be coincident.

Before formalizing this, let us recall that a square integer matrix is called *unimodular* if its determinant is equal to ±1. A *totally unimodular matrix* is a matrix for which every square non-singular sub-matrix is unimodular.

Proposition 13 Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a *bounded discrete* PN system satisfying assumptions (A1) to (A3). If the incidence matrix of the unobservable subnet is *totally unimodular* and the observed transitions fire in integer amounts, then the set $\overline{Y}(\boldsymbol{m}_0, w)$ computed using Algorithm 7 is an integer convex polytope. Additionally, the diagnosis states of the underlying discrete net can be computed using LPPs (10).

Proof: The above statement can be proved using two basic results in [36].

- The first one claims that, if $\boldsymbol{A}$ is a totally unimodular, then matrix $[\boldsymbol{A} \mid \boldsymbol{I}]$ is totally unimodular as well.
- Concerning the second result, let us consider the polyhedron:

$$Q(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{b}', \boldsymbol{c}, \boldsymbol{c}') = \{\boldsymbol{x} \mid \boldsymbol{b} \leq \boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{b}' \text{ and } \boldsymbol{c} \leq \boldsymbol{x} \leq \boldsymbol{c}'\}$$

  where $\boldsymbol{A}$ is a square matrix of integer numbers, and the entries of vectors $\boldsymbol{b}, \boldsymbol{b}', \boldsymbol{c}, \boldsymbol{c}'$ are either integer numbers or $\pm\infty$. Theorem 2 in [36] states that $Q(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{b}', \boldsymbol{c}, \boldsymbol{c}')$ is an integer polyhedron iff $\boldsymbol{A}$ is totally unimodular.

Based on the result in the first item above, we can conclude that, since the incidence matrix of the unobservable subnet $\boldsymbol{C}_u$ is totally unimodular, then the matrix $[\boldsymbol{I} \ \ -\boldsymbol{C}_u]$ is totally unimodular as well.

We now prove that $\overline{Y}(\boldsymbol{m}_0, w)$ is an integer polytope by induction on the length of the observed word.

Basis step: Let us consider the polytope computed in Step 2 of Algorithm 7, namely $\overline{Y}(\boldsymbol{m}_0, \varepsilon)$. The set of constraints defining it can be rewritten as:

$$\begin{cases} [\boldsymbol{I} \;\; -\boldsymbol{C}_u] \cdot \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\sigma}_u \end{bmatrix} = \boldsymbol{m}_0 \\ \boldsymbol{0} \leq \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\sigma}_u \end{bmatrix} \leq \begin{bmatrix} \infty \cdot \boldsymbol{1} \\ \boldsymbol{EN} \end{bmatrix} \end{cases} \tag{11}$$

Now, let

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\sigma}_u \end{bmatrix}, \qquad \boldsymbol{A} = [\boldsymbol{I} \;\; -\boldsymbol{C}_u],$$

$$\boldsymbol{b} = \boldsymbol{b}' = \boldsymbol{m}_0, \qquad \boldsymbol{c} = \boldsymbol{0} \qquad \text{and} \qquad \boldsymbol{c}' = \begin{bmatrix} \infty \cdot \boldsymbol{1} \\ \boldsymbol{EN} \end{bmatrix}.$$

Since $\boldsymbol{A}$ is a unimodular matrix, based on the result in [36] recalled in the second item above, (11) defines an integer polyhedron. Moreover, being $\boldsymbol{0} \leq \boldsymbol{\sigma}_u \leq \boldsymbol{EN}$ and the net system bounded by assumption, all variables are bounded, therefore (11) corresponds to an integer polytope, thus proving the basis step.

Inductive step: Assume that $\overline{Y}(\boldsymbol{m}_0, v)$ is an integer polytope. We want to prove that $\overline{Y}(\boldsymbol{m}_0, vt(\alpha))$ is an integer polytope for any observable transition t and any integer amount α.

Let us preliminary observe that, since $\overline{Y}(\boldsymbol{m}_0, v)$ is an integer polytope by assumption, vectors $\tilde{\boldsymbol{m}}$ and $\tilde{\boldsymbol{\varrho}}$ in (9) have integer entries. Moreover, by the second constraint of (9), it is $\boldsymbol{\sigma}_u = \boldsymbol{\varrho} - \tilde{\boldsymbol{\varrho}}$.

Moreover, let us observe that the set of constraints (9) can be rewritten as:

$$\begin{cases} [\boldsymbol{I} \;\; -\boldsymbol{C}_u] \cdot \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\varrho} \end{bmatrix} = \tilde{\boldsymbol{m}} + \alpha \cdot \boldsymbol{C}(\cdot, t) - \boldsymbol{C}_u \cdot \tilde{\boldsymbol{\varrho}} \\ \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{\varrho}} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\varrho} \end{bmatrix} \leq \begin{bmatrix} \infty \cdot \boldsymbol{1} \\ \boldsymbol{EN} + \tilde{\boldsymbol{\varrho}} \end{bmatrix} \end{cases} \tag{12}$$

Now, let

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{\varrho} \end{bmatrix}, \qquad \boldsymbol{A} = [\boldsymbol{I} \;\; -\boldsymbol{C}_u],$$

$$\boldsymbol{b} = \boldsymbol{b}' = \tilde{\boldsymbol{m}} + \alpha \cdot \boldsymbol{C}(\cdot, t) - \boldsymbol{C}_u \cdot \tilde{\boldsymbol{\varrho}},$$

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{\varrho}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{c}' = \begin{bmatrix} \infty \cdot \boldsymbol{1} \\ \boldsymbol{EN} + \tilde{\boldsymbol{\varrho}} \end{bmatrix}.$$

Since $\boldsymbol{A}$ is a unimodular matrix, based on the result in [36] recalled in the second item above, it follows that (12) is an integer polyhedron. Moreover, since $\boldsymbol{x}$ is bounded being $\tilde{\boldsymbol{\varrho}} \leq \boldsymbol{\varrho} \leq \boldsymbol{EN} + \tilde{\boldsymbol{\varrho}}$ and the net system is bounded, (12) corresponds to an integer polytope. This concludes the proof.

□

There exist algorithms to check total unimodularity of a matrix in polynomial time [37]. Moreover, if the unobservable subnet is either a *state machine* or a *marked graph* this is always true [1] and the set of integer points of the set of consistent markings in the continuous net coincides with the set of consistent markings in the discrete one. Hence, the fault diagnosis approach presented in this paper guarantees to compute the same diagnosis state we obtain using a discrete approach.

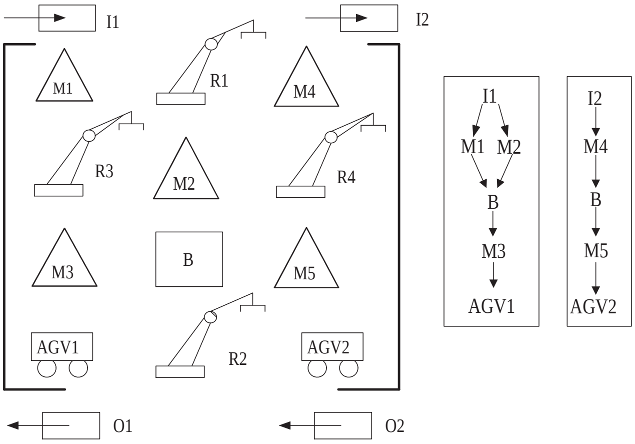

Figure 2: Layout of the automated manufacturing system in Subsection 7.1.

7 Manufacturing examples

In this section we apply the proposed approach to two manufacturing systems. In the first case the unobservable subnet is cyclic, thus it can only be dealt in the continuous framework. In the second case we consider a Petri net whose unobservable subnet is acyclic, thus it can also be dealt in the discrete case. A detailed comparison among the proposed approach and the approach in [19] is presented in terms of computational complexity, and it is shown that, as expected, the advantage of fluidization highly depends on the initial marking of the net. In more detail, it highly increases as the number of reachable markings increases.

7.1 The unobservable subnet is cyclic

We now apply the above approach to a classical automated manufacturing system whose layout is sketched in Fig. 2 and whose Petri net model is shown in Fig. 3. Note that such an example has not been taken from the industrial world. However, it is recognized to be significant in the literature since slight variations of it have already been considered by Zhou and DiCesare in [38], by Basile *et al.* in [39] and by Cabasino *et al.* in [40]. Note however, that while in [38–40] the manufacturing system has been modeled as a discrete Petri net, we now consider the untimed CPN model resulting from the fluidization of the discrete model in [39].

The plant consists of five machines (M1 to M5), four robots (R1 to R4), a finite capacity buffer B, two inputs of raw parts (I1 and I2) of type 1 and type 2 respectively, two Automated Guided Vehicle (AGV) systems (AGV1 and AGV2), and finally two outputs (O1 and O2) for the processed parts. The plant produces two different types of products from two types of raw materials. An unlimited source of raw parts is assumed. It is supposed that there are 19 pallets for the first production line and 20 pallets for the second production line.

This net has $m = 35$ places and $n = 24$ transitions. The marking of place p_{33}, the co-buffer, represents the number of free buffer slots, while the marking

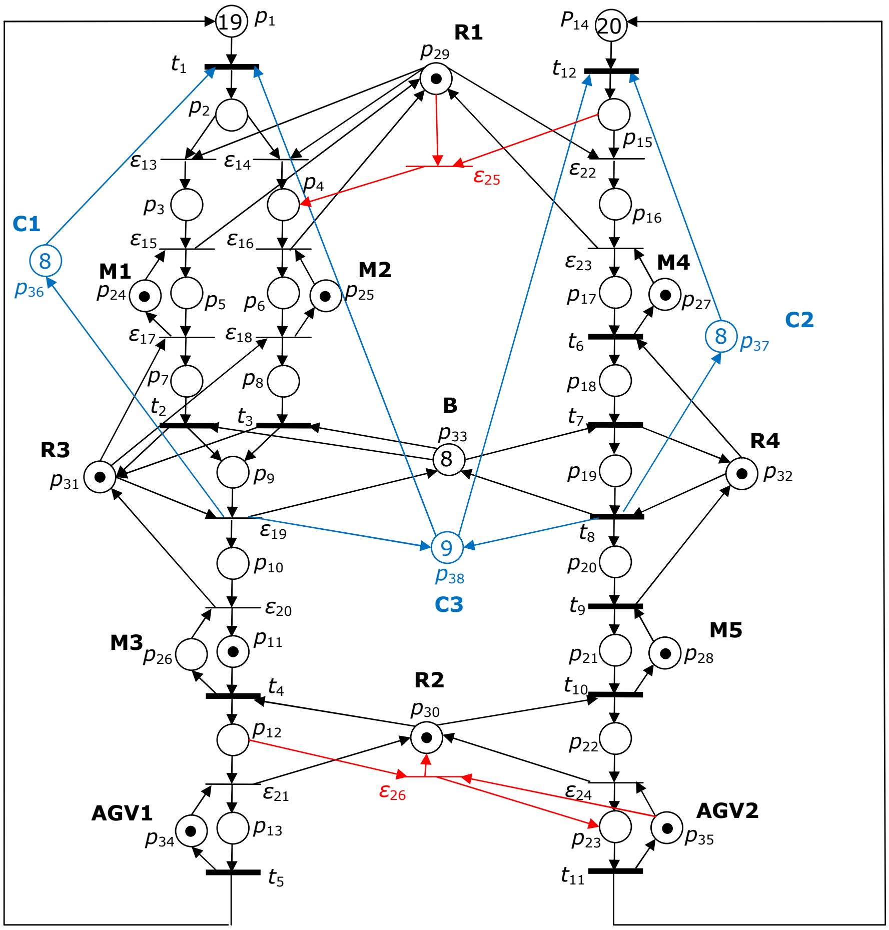

Figure 3: Petri net model of the manufacturing system in Fig. 2.

of places p_9 and p_{19} represent respectively the number of type 1 and type 2 parts present in the buffer. Moreover, there exist 14 circuits.

As in [39], we assume that the system is controlled with the addition of three monitor places (p_{36}, p_{37}, p_{38}) that impose the satisfaction of three Generalized Mutual Exclusion Constraints (GMECs) [41, 42]:

$$\left\{ \begin{array}{ll} \sum_{i=2}^{9} m_i \leq 8 & (a) \\ \sum_{i=15}^{19} m_i \leq 8 & (b) \\ \sum_{i=2}^{9} m_i + \sum_{i=15}^{19} m_i \leq 9 & (c) \end{array} \right. \tag{13}$$

We assume that transitions t_1 to t_{12} correspond to observable events, while transitions ε_{13} to ε_{24} correspond to unobservable but regular events. In particular, we observe the introduction of parts in one of the two production lines (transitions t_1 and t_{12}), the introduction of parts in the buffer by R3 (transitions t_2 and t_3), all operations performed by robot R4 (transitions t_6, t_7, t_8 and t_9), the drawing of parts from one of the two production lines by robot R2 (transitions t_4 and t_{10}) and the output of parts in the AGV systems AGV1 and AGV2 (transitions t_5 and t_{11}).

Finally, we consider two different types of fault modeled by the unobservable transitions ε_{25} and ε_{26}. In particular we assume $T_f^1 = \{\varepsilon_{25}\}$ and $T_f^2 = \{\varepsilon_{26}\}$. The first kind of fault corresponds to a malfunctioning of robot R1 that moves one raw part of the second type to the first production line, so that it is processed by machine M2 instead of M4. The second kind of fault corresponds to a malfunctioning of robot R2 that moves one part of the first type, after it has been processed by machine M3, and sends it to AGV2 who directs it to the wrong output (O2 instead of O1). Note that using fluidization is a requirement here being the unobservable net cyclic (see e.g. the cycles $\varepsilon_{13}p_3\varepsilon_{15}p_{29}$, $\varepsilon_{16}p_6\varepsilon_{18}p_{25}$, and so on).

Now, let us assume that the word

$$\begin{array}{rcl} w & = & t_1(1)t_1(1)t_2(1)t_{12}(1)t_3(1)t_{12}(1)t_3(1)t_6(1) \\ & & t_7(1)t_8(1)t_4(1)t_5(1)t_9(1)t_{10}(1)t_{11}(1) \end{array}$$

is observed. The results of computations carried out on a PC Intel with a clock of 1.80 GHz, are briefly summarized in Table 1. In particular, here we reported: the number of vertices N_v of the set $\overline{Y}(v, \boldsymbol{m}_0)$ for all prefixes v of w, the time T_v necessary to compute them, and the corresponding diagnosis states. The MATLAB software used for computation is available on-line [35]. Note that for simplicity of notation in Table 1 we omitted the amount of firing of the observations, that are unitary in all cases.

Let us also observe that the times to compute the diagnosis states, once the set of vertices is given, is omitted here because in all cases it is practically negligible (less than one second).

Moreover, let us observe that in this case, as it occurs in general cases, the number of vertices is not related to the length of the observed word. What is happening here is that the time to compute the diagnosis state at a given observation increases when the number of vertices at the previous observation increases. This is a direct consequence of the algorithm used to compute them.

For the sake of brevity we do not report all vertices. As an example, in Table 2 we only summarize the set of vertices of $\overline{Y}(t_1(1), \boldsymbol{m}_0)$ that includes 7 entries, namely $\boldsymbol{e}_1$ to $\boldsymbol{e}_7$. In particular, these vertices correspond respectively, to the firing of the following unobservable sequences at the marking reached

Observed word v	N_v	T_v [sec]	$\Delta(v, T_f^1)$	$\Delta(v, T_f^2)$
ε	1	7.67	N	N
t_1	7	0.81	N	N
t_1t_1	20	3.33	N	N
$t_1t_1t_2$	12	2.72	N	N
$t_1t_1t_2t_{12}$	51	5.92	U	N
$t_1t_1t_2t_{12}t_3$	18	4.70	U	N
$t_1t_1t_2t_{12}t_3t_{12}$	59	10.63	U	N
$t_1t_1t_2t_{12}t_3t_{12}t_3$	18	5.27	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6$	2	1.25	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7$	2	1.25	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8$	2	1.16	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8t_4$	12	4.63	F	U
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8t_4t_5$	4	5.58	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8t_4t_5t_9$	4	5.31	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8t_4t_5t_9t_{10}$	8	8.09	F	N
$t_1t_1t_2t_{12}t_3t_{12}t_3t_6t_7t_8t_4t_5t_9t_{10}t_{11}$	4	5.05	F	N

Table 1: Results of some numerical simulations carried out on the untimed CPN system in Fig. 3.

from $\boldsymbol{m}_0$ firing t_1 for a unitary amount:

$$\begin{array}{ll}
\sigma_u^{(1)} = \varepsilon_{14}(1)\varepsilon_{16}(1)\varepsilon_{18}(1), & \sigma_u^{(2)} = \varepsilon_{13}(1)\varepsilon_{15}(1)\varepsilon_{17}(1),\\
\sigma_u^{(3)} = \varepsilon_{14}(1)\varepsilon_{16}(1), & \sigma_u^{(4)} = \varepsilon_{13}(1)\varepsilon_{15}(1),\\
\sigma_u^{(5)} = \varepsilon_{14}(1), & \sigma_u^{(6)} = \varepsilon_{13}(1),\\
\sigma_u^{(7)} = \varepsilon. &
\end{array}$$

Clearly, no fault occurrence may have been occurred when the observation is $t_1(1)$ thus the two diagnosis states are both equal to N.

The first uncertain state occurs after the observation of $w = t_1(1)t_1(1)t_2(1)t_{12}(1)$. This is consistent with the fact that there exist sequences, such as

$$w' = t_1(1)t_1(1)\varepsilon_{13}(1)\varepsilon_{15}(1)\varepsilon_{17}(1)t_2(1)t_{12}(1)\varepsilon_{25}(1),$$

that are consistent with the observation and contain the fault transition ε_{25}, but there also exist sequences consistent with the observation that do not contain it, such as

$$w'' = t_1(1)t_1(1)\varepsilon_{13}(1)\varepsilon_{15}(1)\varepsilon_{17}(1)t_2(1)t_{12}(1).$$

On the contrary, the diagnosis state relative to the first fault class is equal to F after the observation $w = t_1(1)t_1(1)t_2(1)t_{12}(1)t_3(1)t_{12}(1)t_3(1)$. The correctness of this is evident. In fact, given the initial marking, if t_1 has been observed for an amount 2 the total amount of firing of t_2 plus t_3 may be greater than 2 if and only if transition ε_{25} has fired.

Similar considerations can be repeated to explain the other diagnosis states.

	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$	$\boldsymbol{e}_5$	$\boldsymbol{e}_6$	$\boldsymbol{e}_7$
p_1	18	18	18	18	18	18	18
p_2	0	0	0	0	0	0	1
p_3	0	0	0	0	0	1	0
p_4	0	0	0	0	1	0	0
p_5	0	0	0	1	0	0	0
p_6	0	0	1	0	0	0	0
p_7	0	1	0	0	0	0	0
p_8	1	0	0	0	0	0	0
p_9	0	0	0	0	0	0	0
p_{10}	0	0	0	0	0	0	0
p_{11}	1	1	1	1	1	1	1
p_{12}	0	0	0	0	0	0	0
p_{13}	0	0	0	0	0	0	0
p_{14}	20	20	20	20	20	20	20
p_{15}	0	0	0	0	0	0	0
p_{16}	0	0	0	0	0	0	0
p_{17}	0	0	0	0	0	0	0
p_{18}	0	0	0	0	0	0	0
p_{19}	0	0	0	0	0	0	0
p_{20}	0	0	0	0	0	0	0
p_{21}	0	0	0	0	0	0	0
p_{22}	0	0	0	0	0	0	0
p_{23}	0	0	0	0	0	0	0
p_{24}	1	1	1	0	1	1	1
p_{25}	1	1	0	1	1	1	1
p_{26}	0	0	0	0	0	0	0
p_{27}	1	1	1	1	1	1	1
p_{28}	1	1	1	1	1	1	1
p_{29}	1	1	1	1	0	0	1
p_{30}	1	1	1	1	1	1	1
p_{31}	0	0	1	1	1	1	1
p_{32}	1	1	1	1	1	1	1
p_{33}	8	8	8	8	8	8	8
p_{34}	1	1	1	1	1	1	1
p_{35}	1	1	1	1	1	1	1
p_{36}	7	7	7	7	7	7	7
p_{37}	8	8	8	8	8	8	8
p_{38}	8	8	8	8	8	8	8

	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$	$\boldsymbol{e}_5$	$\boldsymbol{e}_6$	$\boldsymbol{e}_7$
ε_{13}	0	1	0	1	0	1	0
ε_{14}	1	0	1	0	1	0	0
ε_{15}	0	1	0	1	0	0	0
ε_{16}	1	0	1	0	0	0	0
ε_{17}	0	1	0	0	0	0	0
ε_{18}	1	0	0	0	0	0	0
ε_{19}	0	0	0	0	0	0	0
ε_{20}	0	0	0	0	0	0	0
ε_{21}	0	0	0	0	0	0	0
ε_{22}	0	0	0	0	0	0	0
ε_{23}	0	0	0	0	0	0	0
ε_{24}	0	0	0	0	0	0	0
ε_{25}	0	0	0	0	0	0	0
ε_{26}	0	0	0	0	0	0	0

Table 2: The set of vertices (in column) of $\overline{Y}(t_1(1), \boldsymbol{m}_0)$.

7.2 The unobservable subnet is acyclic

The second example we consider is voluntarily simple. It aims to show that, even in very simple examples with no unobservable cycle, it may happen that the discrete approach fails due to the exponential growth of the number of basis markings, and a fortiori of the number of reachable states, while the continuous approach reveals efficient due to a small number of vertices.

Let us consider the Petri net in Fig. 4. It represents a part of a large manufacturing system consisting of several machines, robots and buffers. In particular, transitions t_1 and t_2 model two robots R_1 and R_2, that take parts from two different buffers modeled by places p_1 and p_2, respectively. The four parts taken by robot R_1 are packed in couples and placed on two different conveyor belts modeled respectively by places p_3 and p_4, that follow two parallel lines at two different levels. In more detail, p_4 is located in the lowest level, while p_3 is in the highest level.

Parts in the conveyor belts represented by place p_4 are processed by the machine modeled by transition ε_7 and then put in a common buffer represented by place p_7.

The bottom part of the net models similar operations.

Transitions $t_3 and t_4$ model respectively the output of parts from the conveyor belts modeled by p_3 and p_6 to a common buffer modeled by p_8, while transition t_5 models the output of parts from the common buffer p_7. To each part exiting p_7 and p_8 corresponds a new part entering p_1 and a new part entering p_2, and the process repeats cyclically.

As usual, transitions t_j, $j = 1, \ldots, 6$, represent observable transitions, while transitions ε_i, $i = 7, \ldots, 10$ model silent transitions. In more detail, ε_7 and ε_8 represent regular events, while ε_9 and ε_{10} model fault events, i.e., some breakage in the highest conveyor belts at the beginning of the two main production lines. Finally, we assume that the two fault transitions belong to two different fault classes, i.e., $T_f^1 = \{\varepsilon_9\}$ and $T_f^2 = \{\varepsilon_{10}\}$.

Our goal here is that of evaluating the effectiveness of fluidization with respect to fault diagnosis. To this aim, we apply to the above example both the discrete approach in [19] and the continuous approach proposed in this paper, and provide a comparison among them in terms of computational complexity.

The diagnosis approach in [19] is based on the notion of basis markings, that are a subset of the set of consistent markings. In particular, given an observed word w, a basis marking is a marking that has been reached firing w and all those unobservable transitions that are *strictly* necessary to enable it. The number of basis markings clearly affects the computational complexity, since they need to be exhaustively enumerated, as well as the number of vertices of $\overline{Y}(w, \boldsymbol{m}_0)$ in the continuous case.

Note that, as well as in the previous example, numerical simulations are carried out on a PC Intel with a clock of 1.80 GHz. Moreover, the discrete approach is implemented using the MATLAB tool in [43].

Two different scenarios are considered in the discrete case. First, we assume as initial marking $\boldsymbol{m}_0' = [20\ 20\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and as observed word $w' = t_1 t_2 t_5 t_5 t_5 t_1 t_3 t_4$.

Secondly, we assume $\boldsymbol{m}_0'' = [80\ 80\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and

$w'' = t_1 t_1 t_1 t_1 t_2 t_2 t_2 t_2 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_5 t_1 t_1 t_1 t_1 t_3 t_3\ t_3 t_3 t_4 t_4 t_4 t_4$.

The resulting number N_{M_b} of basis markings, the time T_{M_b} necessary to

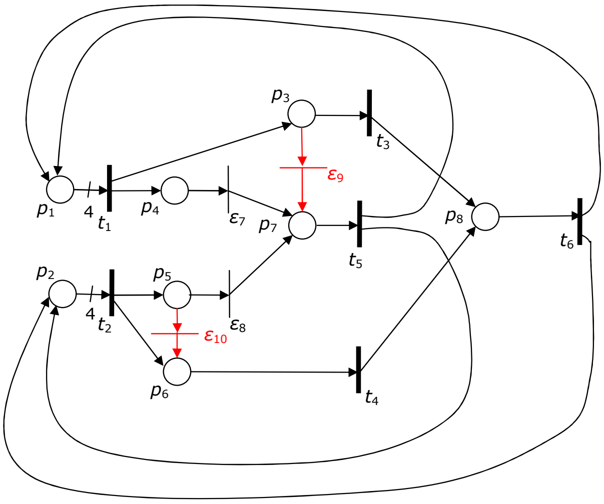

Figure 4: The Petri net considered in Subsection 7.2.

Discr. Observed word v	N_{M_b}	T_{M_b} $[sec]$	$\Delta(v,T_f^1)$, $\Delta(v,T_f^2)$
ε	1	0.335	N, N
t_1	1	0.226	U, N
t_1t_2	1	0.044	U, U
$t_1t_2t_5$	3	0.080	U, U
$t_1t_2t_5t_5$	6	0.038	U, U
$t_1t_2t_5t_5t_5$	6	0.052	F, N
$t_1t_2t_5t_5t_5t_4$	6	0.027	F, N
$t_1t_2t_5t_5t_5t_4t_1$	6	0.051	F, N
$t_1t_2t_5t_5t_5t_4t_1t_3$	6	0.036	F, N

Table 3: Results of some numerical simulations carried out on the PN in Fig. 4 assuming $\boldsymbol{m}_0 = [20\ 20\ 0\ 0\ 0\ 0\ 0]^T$.

compute them and the diagnosis states are reported in Tables 3 and Table 4, respectively.

Both scenarios can be simulated in the continuous case assuming as initial marking $\boldsymbol{m}_0''' = [2\ 2\ 0\ 0\ 0\ 0\ 0]^T$ and observed word

$$w''' = t_1(0.1)t_2(0.1)t_5(0.3)t_4(0.1)t_1(0.1)t_3(0.1).$$

In particular, in the first case fluidization assumes that 10 discrete tokens correspond to a unit of fluid content in the continuous PN system, while in the second case 40 discrete tokens are approximated by a unit of fluid content.

The resulting number of vertices N_v of $\overline{Y}(w, \boldsymbol{m}_0)$ and the time T_v to compute them are reported in Table 5 where the last column also shows the diagnosis states for the two fault classes. As it can be observed the diagnosis states computed in the continuous case are in accordance with the discrete ones.

The advantages in terms of computational complexity are quite negligible

Discr. Observed word v	N_{M_b}	T_{M_b} $[sec]$	$\Delta(v, T_f^1)$, $\Delta(v, T_f^2)$
ε	1	0.255	N, N
$t_1t_1t_1t_1$	1	0.149	U, N
$t_1t_1t_1t_1t_2t_2t_2t_2$	1	5.790	U, U
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5$	81	18.100	U, U
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5t_5t_5t_5t_5$	4830	43.463	U, U
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5$	34650	297.308	F, N
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_4t_4t_4t_4$	34650	227.509	F, N
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_4t_4t_4t_4t_1t_1t_1t_1$	34650	739.524	F, N
$t_1t_1t_1t_1t_2t_2t_2t_2t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_5t_4t_4t_4t_4t_1t_1t_1t_1t_3t_3t_3t_3$	34650	266.386	F, N

Table 4: Results of some numerical simulations carried out on the PN in Fig. 4 assuming $\boldsymbol{m}_0 = [80\ 80\ 0\ 0\ 0\ 0\ 0]^T$.

Cont. Observed word v	N_v	T_v $[sec]$	$\Delta(v, T_f^1)$, $\Delta(v, T_f^2)$
ε	1	0.01	N, N
$t_1(0.1)$	4	0.06	U, N
$t_1(0.1)t_2(0.1)$	12	0.04	U, U
$t_1(0.1)t_2(0.1)t_5(0.3)$	1	0.03	F, N
$t_1(0.1)t_2(0.1)t_5(0.3)t_4(0.1)$	1	0.02	F, N
$t_1(0.1)t_2(0.1)t_5(0.3)t_4(0.1)t_1(0.1)$	4	0.04	F, N
$t_1(0.1)t_2(0.1)t_5(0.3)t_4(0.1)t_1(0.1)t_3(0.1)$	2	0.04	F, N

Table 5: Results of some numerical simulations carried out on the CPN system obtained from the fluidization of the PN in Fig. 4 assuming $\boldsymbol{m}_0 = [2\ 2\ 0\ 0\ 0\ 0\ 0]^T$.

in the case of the first discrete scenario, while they become evident in the case of the second scenario. Such advantages become even more significant if we consider the same discrete PN system with an even larger number of reachable states, e.g. the one obtained multiplying $\boldsymbol{m}_0'''$ by 50. In particular, in such a case the simulation does not end after one day.

Summarizing, we conclude that the advantages of fluidization depend on the considered net system, and in general there is no a priori relationship between the number of vertices of the set $\overline{Y}(w, \boldsymbol{m}_0)$ and the number of basis markings. This depends on the structure of the unobservable subnet, on the initial marking and on the observed word. Nevertheless, as intuitive, major advantages are in general obtained when the number of reachable markings in the discrete case is large.

8 Conclusions

In this paper we investigated the effect of fluidization of Petri nets with respect to fault diagnosis. In particular the focus is on untimed continuous Petri nets. Two are the main conclusions of such research.

The first one is that fluidization allows to relax the assumption, common to all the discrete event system diagnosis approaches, that there exist no cycle of unobservable transitions.

The second one is that there may exist cases where fluidization leads to significant advantages in terms of computational complexity, enabling us to also perform diagnosis on systems whose number of reachable states is so large that discrete approaches are not applicable in practice. A very simple case of this is given in the paper.

In the next future we plan to study the problem of diagnosability of untimed continuous Petri nets, i.e., determine some criteria to establish a priori if fault occurrences can be reconstructed after a finite amount of observations.

References

[1] C. Mahulea, C. Seatzu, M. P. Cabasino, L. Recalde, and M. Silva, "Observer Design for Untimed Continuous Petri Nets," in *American Control Conference*, St. Louis, Missouri, USA, June 2009, pp. 4765 – 4770.

[2] C. Seatzu, M. P. Cabasino, C. Mahulea, and M. Silva, "New results for fault detection of untimed continuous Petri nets," in *48th IEEE Conference on Decision and Control*, Shangai, China, December 2009, pp. 6952–6957.

[3] C. Seatzu, C. Mahulea, M. P. Cabasino, and M. Silva, "Fault diagnoser design for untimed continuous Petri nets," in *3rd IEEE Multi-Conference on Systems and Control*, Saint Petersburg, Russia, July 2009, pp. 1598–1604.

[4] A. Rosich, E. Frisk, J. Aslund, R. Sarrate, and F. Nejjari, "Fault Diagnosis Based on Causal Computations," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011, in press.

[5] D. Lefebvre and E. Leclercq, "Stochastic Petri Net Identification for the Fault Detection and Isolation of Discrete Event Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 2, pp. 213–225, March 2011.

[6] A. Ramírez-Trevino, E. Ruiz-Beltrán, J. Arámburo-Lizárraga, and E. López-Mellado, "Structural Diagnosability of DES and Design of Reduced Petri Net Diagnosers," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011, in press.

[7] J. Meseguer, V. Puig, and T. Escobet, "Fault Diagnosis Using a Timed Discrete-Event Approach Based on Interval Observers: Application to Sewer Networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 5, pp. 900–916, September 2010.

[8] S. Takai and R. Kumar, "Decentralized Diagnosis for Nonfailures of Discrete Event Systems Using Inference-Based Ambiguity Management," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 2, pp. 406–412, March 2010.

[9] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosis of intermittent faults," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 14, no. 2, pp. 171–202, April 2004.

[10] D. Bertsimas, D. Gamarnik, and J. Tsitsiklis, "Stability conditions for multiclass fluid queueing networks," *IEEE Transactions on Automatic Control*, vol. 41, no. 11, pp. 1618–1631, November 2002.

[11] H. Chen and D. Yao, *Fundamentals of queueing networks: Performance, asymptotics, and optimization.* Springer Verlag, 2001.

[12] G. Sun, C. Cassandras, and C. Panayiotou, "Perturbation analysis of multiclass stochastic fluid models," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 14, no. 3, pp. 267–307, June 2004.

[13] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets.* Springer-Verlag, 2010, 2^{nd} edition.

[14] M. Silva and L. Recalde, "On fluidification of Petri net models: from discrete to hybrid and continuous models," *Annual Reviews in Control*, vol. 28, no. 2, pp. 253–266, December 2004.

[15] J. Júlvez and R. Boel, "A Continuous Petri Net Approach for Model Predictive Control of Traffic Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 4, pp. 686–697, July 2010.

[16] M. Silva, J. Júlvez, C. Mahulea, and C. Vázquez, "On fluidization of discrete event models: observation and control of continuous Petri nets," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 21, no. 4, pp. 1–71, December 2011.

[17] L. Recalde, E. Teruel, and M. Silva, "Autonomous continuous P/T systems," in *Application and Theory of Petri Nets 1999*, ser. Lecture Notes in Computer Science, J. K. S. Donatelli, Ed., vol. 1639. Springer, 1999, pp. 107–126.

[18] S. Genc and S. Lafortune, "Distributed Diagnosis of Place-Bordered Petri Nets," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 2, pp. 206–219, April 2007.

[19] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, September 2010.

[20] M. Dotoli, M. Fanti, A. Mangini, and W. Ukovich, "On-line fault detection in discrete event systems by Petri nets and integer linear programming," *Automatica*, vol. 45, no. 11, pp. 2665–2672, November 2009.

[21] R. Boel and J. van Schuppen, "Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers," in *Proc. WODES'02: 6th Workshop on Discrete Event Systems*, Zaragoza, Spain, October 2002, pp. 175–181.

[22] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete-event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 20, no. 1–2, pp. 33–79, January 2000.

[23] S. H. Zad, R. Kwong, and W. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, July 2003.

[24] S. Jiang and R. Kumar, "Failure diagnosis of discrete-event systems with linear-time temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 934–945, June 2004.

[25] J. Lunze and J. Schroder, "Sensor and actuator fault diagnosis of systems with discrete inputs and outputs," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 3, June 2004.

[26] M. Sampath and S. Lafortune, "Active diagnosis of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 7, pp. 908–929, July 1998.

[27] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, September 1995.

[28] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete event systems, a net unfolding approach," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 714–727, 2003.

[29] G. Jiroveanu and R. Boel, "A distributed approach for fault detection and diagnosis based on time Petri nets," *Mathematics and Computers in Simulation*, vol. 70, no. 5, pp. 287–313, February 2006.

[30] D. Lefebvre and C. Delherm, "Diagnosis of DES with Petri net models," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 31–39, January 2007.

[31] A. Ramirez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel, and E. López-Mellado, "Online fault diagnosis of discrete event systems. A Petri net-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, 2007.

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

[33] J. Júlvez, L. Recalde, and M. Silva, "On reachability in autonomous continuous Petri net systems," in 24^{th} *International Conference on Application and Theory of Petri Nets*, ser. Lecture Notes in Computer Science, W. van der Aalst and E. Best, Eds. Eindhoven, The Netherlands: Springer, 2003, vol. 2679, pp. 221–240.

[34] J. Campos, G. Chiola, and M. Silva, "Ergodicity and throughput bounds of Petri net with unique consistent firing count vector," *IEEE Transactions on Software Engineering*, vol. 17, no. 2, pp. 117–125, February 1991.

[35] C. Mahulea, "Matlab toolbox for the diagnosis of ContPNs," http://webdiis.unizar.es/∼cmahulea/research/diagnoserContPN.zip, 2009.

[36] A. Hoffman and J. Kruskal, "Integral boundary points of convex polyhedra," in *Linear Inequalities and Related Systems. Annals of Mathematics Studies*, H. Kuhn and A. Tucker, Eds. Princeton University Press, 1956, vol. 38, pp. 223–246.

[37] K. Truemper, "A decomposition theory for matroids. V. Testing of matrix total unimodularity," *Journal of Combinatorial Theory, Series B*, vol. 49, no. 2, pp. 241 – 281, August 1990.

[38] M. Zhou and F. DiCesare, *Petri net synthesis for discrete event control of manufacturing systems.* Kluwer, 1993.

[39] F. Basile, A. Giua, and C. Seatzu, "Petri net control using event observers and timing information," in *Proc. 41th IEEE Conf. on Decision and Control*, Las Vegas, USA, December 2002, pp. 787–792.

[40] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Engineering Practice*, vol. 19, no. 9, pp. 989–1001, September 2011.

[41] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in Proc. 1992 IEEE Int. Conf. on Systems, Man, and Cybernetics, Chicago, USA, October 1992, pp. 974 – 979.

[42] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, January 1996.

[43] M. Pocci, "Matlab toolbox for the diagnosis of discrete PNs," http://www.diee.unica.it/giua/TESI/09_Marco.Pocci/PN_DIAG.zip, 2009.

This appendix summarizes the main notations used in the paper.

- P: set of places;
- T: set of transitions;
- T_o (T_u, T_{reg}, T_f): set of observable (unobservable, regular, faulty) transitions;
- T_f^i: the i-th fault class;
- $\boldsymbol{Pre}$ ($\boldsymbol{Post}$): pre (post) incidence matrix;
- $\boldsymbol{C}$: incidence matrix;
- $\boldsymbol{C}_o$ ($\boldsymbol{C}_u$): restriction of $\boldsymbol{C}$ to T_o (T_u);
- $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$: net structure;
- $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$: net system with initial marking $\boldsymbol{m}_0$;
- $\mathcal{L}(\mathcal{N}, \boldsymbol{m}_0)$: set of firable sequences at $\boldsymbol{m}_0$;
- $\mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$: set of markings that are reachable with a finite firing sequence at $\boldsymbol{m}_0$;
- ${}^\bullet x$ ($x^\bullet$): input (output) set of a node $x \in P \cup T$;
- Π: projection operator;
- $\mathcal{L}(w)$: set of firing sequences consistent with the observed word w;
- $\Gamma(w)$: set of unobservable sequences consistent with the observed word w;
- $\mathcal{C}(w)$: set of markings consistent with the observed word w;
- $\overline{Y}(\boldsymbol{m}_0, w)$: set of $\boldsymbol{y}$-vectors associated to $\boldsymbol{m}_0$ and the observed word w (see (5));
- $Y(\boldsymbol{m}_0, w)$: set of ϱ-vectors associated to $\boldsymbol{m}_0$ and the observed word w (see (6));
- $EN(t)$: structural enabling bound of transition t (see (8));
- $\Delta(w, T_f^i)$: the diagnosis state relative to the observed word w and the fault class T_f^i.