Fault diagnosis in labelled Petri nets

Al-Ajeli, Ahmed ; Parker, David

DOI:
10.1016/j.automatica.2021.109831

Automatica 132 (2021) 109831

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Fault diagnosis in labelled Petri nets: A Fourier–Motzkin based approach ☆

Ahmed Al-Ajeli [a,*], David Parker [b]

[a] College of Information Technology, University of Babylon, Babylon, Iraq
[b] School of Computer Science, University of Birmingham, Birmingham, UK

ARTICLE INFO

Article history:
Received 7 December 2019
Received in revised form 17 February 2021
Accepted 9 June 2021
Available online xxxx

Keywords:
Discrete-event systems
Petri nets
Fault diagnosis
Fourier–Motzkin elimination

ABSTRACT

We propose techniques for fault diagnosis in discrete-event systems modelled by labelled Petri nets, where fault events are modelled as unobservable transitions. The proposed approach combines an offline and an online algorithm. The offline algorithm constructs a diagnoser in the form of sets of inequalities that capture the legal, normal and faulty behaviour. To implement the offline algorithm, we adopt the Fourier–Motzkin method for elimination of variables from these sets of inequalities. Upon observing an event, the diagnoser is used to determine whether a fault occurred or might have occurred. The occurrence of a fault can be verified by checking the observed sequence against the sets of inequalities. This approach has the advantage that the tradeoff between the size of the diagnoser and the time for computing the diagnosis is achieved. In addition, fault diagnosis in both bounded and unbounded Petri nets can be addressed.

1. Introduction

The safety and reliability of large complex systems play an important role in the availability of the services provided by them. Unfortunately, fault occurrences in such systems are usually unavoidable. Fault diagnosis addresses the problem of detecting and isolating these fault occurrences. Thus, developing automatic approaches to obtain accurate and timely diagnosis decisions in such systems enhances their safety and reliability. It is well known that the problem of fault diagnosis in partially-observed discrete-event systems (DES) is a complex problem; it has been studied by many researchers in order to develop methods in which the time and the space complexity are balanced.

The traditional approach to solving this problem is by assuming that there is a model capturing the behaviour of the system to be diagnosed (also called the plant). Two formalisms are usually used in the literature: automata and Petri nets (Basile et al., 2008; Cabasino et al., 2010; Dotoli et al., 2009; Sampath et al., 1995). In this formalism, faults are modelled as unobservable events. The problem of fault diagnosis under partial observation was first investigated by Sampath et al. (1995). The authors modelled the system behaviour as a regular language captured by an automaton and the solution starts by creating, from this model, an automaton called a diagnoser in which all events are observable. One of the limitations of this approach, however, is the inability to handle infinite systems (i.e., unbounded state spaces).

Petri net models provide more attractive graphical and mathematical features which can be used for the purpose of dealing with both finite and infinite systems. An extension to the idea introduced in the automata context has been proposed for Petri nets (Cabasino et al., 2010; Jiroveanu et al., 2008; Zhu et al., 2018). The aim was to reduce the computational cost by only enumerating a subset of the reachable markings in the system being diagnosed.

A different idea has been proposed in Basile et al. (2009) and Dotoli et al. (2009), where they use equations to address the diagnosis problem, rather than representing the diagnoser as an automaton. More specifically, the fault diagnosis problem has been reduced to an integer linear programming (ILP) problem, which is solved online every time an event is observed. Using this idea, the space complexity is reduced at the cost of the time complexity, which could be exponential. For a review of approaches for fault diagnosis in DES, we refer the reader to Basile (2014), Cabasino et al. (2012) and Zaytoon and Lafortune (2013).

The above contributions have been demonstrated in the context of Petri nets where no two transitions in the model of the system share the same label. Extensions to the work of Cabasino et al. (2011) and Fanti et al. (2013) have been reported in Cabasino et al. (2010), Dotoli et al. (2009) and Wang et al.

☆ The material in this paper was partially presented at the 12th UKACC International Conference on Control, September 6–7, 2018, Sheffield, UK. This paper was recommended for publication in revised form by Associate Editor Prashant Mhaskar under the direction of Editor Thomas Parisini.

* Corresponding author.
E-mail addresses: a.alajeli@uobabylon.edu.iq (A. Al-Ajeli), d.a.parker@cs.bham.ac.uk (D. Parker).

https://doi.org/10.1016/j.automatica.2021.109831
0005-1098/

(2020) to the cases of *labelled* Petri nets (LPN) in which there is no restriction on having unique labels associated with transitions. These transitions can be simultaneously enabled (indistinguishable transitions), but only one of them can fire. In addition, Basile et al. proposed an approach for both diagnosability and fault detection in labelled Petri nets exploiting the ILP approach (Basile et al., 2012). Recently, a diagnostic technique using an online count vector estimation was designed (Chouchane et al., 2020; Zhu et al., 2020). These techniques are based on solving a fewer number of LP problems for an observed sequence of events.

Alternatively, a new approach adopting the idea of variable elimination from a set of inequalities has been developed for fault diagnosis in Petri nets (Al-Ajeli & Bordbar, 2016; Al-Ajeli & Parker, 2018). The integer Fourier–Motzkin elimination method (IFME) has been used for the elimination (Pugh, 1991; Williams, 1976). IFME is an extension of the Fourier–Motzkin elimination (FME) method used for inequalities in real variables (Conforti et al., 2014; Duffin, 1974; Kohler, 1967).

In this paper, we further extend the previous work based on the IFME method to the case of labelled Petri nets under the assumption that observable transitions might be indistinguishable. The proposed solution is in two parts: offline and online. The diagnoser is constructed offline as sets of inequalities. During the online step, a sequence of observed events (labels) is obtained and verified against the sets of inequalities constructed in the offline step to make the diagnosis decisions. It is worth mentioning that the present approach does not use the IFME method for solving an ILP problem, neither online nor offline. Instead, the method is used for the purpose of projecting the space described by a set of inequalities by eliminating variables.

This paper is structured as follows. In Section 2, a general background of Petri nets and the IFME method is provided. Section 3 presents a description of the fault diagnosis problem in DES. The details of the proposed approach and a proof of correctness for this approach on the fault diagnosis problem are covered in Section 4. Conclusions and future directions are discussed in Section 5.

2. Background

2.1. Petri nets

A *Petri net* (Murata, 1989) is defined as a four tuple $\mathcal{N} = (P, T, Pre, Post)$, where P and T are non-empty finite sets of places and transitions, respectively; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the weights of the arcs from places to transitions and from transitions to places. We use $m = |P|$ and $n = |T|$ for the number of places and transitions. For a given transition $t \in T$, an *input* (resp. *output*) place of t is a place p such that $Pre(p, t)$ (resp. $Post(p, t)$) is positive. $A = Post - Pre$ is the incidence matrix of a net.

A *state* of a Petri net, known as a *marking*, is represented as $M : P \to \mathbb{N}$ capturing the number of tokens in each place. We sometimes represent a marking as an $m \times 1$ matrix of non-negative integers. A transition t is *enabled* at a marking M if $M(p) \geq pre(p, t)$ for each input place p of t. An enabled transition can *fire*, resulting in a new marking M', denoted by $M \xrightarrow{t} M'$. We can find the reachable marking M' by $M' = M + A\mathbf{u}$, where $\mathbf{u}$ is the n-dimensional firing vector of the transition t. A sequence of transitions $\boldsymbol{\sigma} = t_1 \dots t_l$ of T is called *enabled* at a marking M if there are markings $M_1, \dots, M_l$ so that $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \cdots \xrightarrow{t_l} M_l$. In this case, we write $M \xrightarrow{\sigma} M_l$ and refer to M_l as a marking *reachable* from M and σ is the firing sequence. We write $R(\mathcal{N}, M)$ for the set of all markings reachable from M. The initial marking of the system is represented by an *initial marking* M_0. We will write $(\mathcal{N}, M_0)$ for a Petri net with its initial marking M_0.

Suppose that we have a sequence σ of $(\mathcal{N}, M_0)$, then the *Parikh vector* $\# : T^* \to \mathbb{N}^n$ is a map which assigns to every sequence σ a vector $\#(\sigma)$ in which each element represents the number of firings of each transition in σ. In other words, for $\#(\sigma) : T \to \mathbb{N}$, $\#(\sigma)(t)$ is the number of occurrence of $t \in T$ within the sequence σ. Sometimes, we also write $\#(t, \sigma)$ to represent the number of the occurrences of t in σ.

The set of sequences of transitions resulting in reachable markings is called the *language* of the Petri net and is denoted by $L(\mathcal{N}, M_0)$, i.e., $L(\mathcal{N}, M_0) = \{\sigma \mid \exists M\ M_0 \xrightarrow{\sigma} M\}$. Suppose that a destination marking M is reachable from M_0 in a Petri net $\mathcal{N}$ through a sequence σ, we can then find M using the following *state equation*:

$$M = M_0 + A\mathbf{x} \geq \vec{\mathbf{0}} \tag{1}$$

where A is the incidence matrix of $\mathcal{N}$, and $\mathbf{x} \in \mathbb{N}^n$ is an n-dimensional column vector with $\mathbf{x} = (x_1, \dots, x_n)$ and $x_i = \#(t_i, \sigma)$ for $t_i \in T$. Then, for any sequence $\sigma \in L(\mathcal{N}, M_0)$, there exists $\mathbf{x} = \#(\sigma)$ satisfying (1). The converse is not always true. In some cases, e.g. *acyclic* Petri nets, the converse holds too.

Definition 1 (*Tsuji & Murata, 1993*)**.** Let $\nu = (\alpha_1, \dots, \alpha_n)$ be a solution of the state equation for a Petri net $(\mathcal{N}, M_0)$ with a destination marking M. Then, the *firing count subnet* with respect to ν is the subnet $\mathcal{N}_\nu$ where each transition t_i in $\mathcal{N}_\nu$ is such that $\alpha_i > 0$ together with its input and output places and its connecting arcs. $M_{0\nu}$ and M_ν denote the restrictions of M_0 and M to places in $\mathcal{N}_\nu$.

Lemma 1 (*Al-Ajeli & Parker, 2018*)**.** *Suppose that ν is an $n \times 1$ column vector and M is a reachable marking in a Petri net $\mathcal{N}$ such that $M' = M + A\nu \geq \vec{\mathbf{0}}$. Considering that $\mathcal{N}_\nu$ (see Definition 1) is cycle-free, then there exists a sequence $\sigma \in T_\nu^*$ (T_ν is the set of transitions in $\mathcal{N}_\nu$) such that $M_\nu \xrightarrow{\sigma} M'_\nu$ and $\#(\sigma) = \nu$, where M_ν and M'_ν are restrictions of M and M' to places of $\mathcal{N}_\nu$. In addition, σ can fire under M resulting in M' such that $M \xrightarrow{\sigma} M'$.*

Now, suppose that we have a Petri net $(\mathcal{N}, M_0)$, then the association of a label $e \in \Sigma$, where Σ represents a set of labels (alphabet), to transitions in $\mathcal{N}$ is called a *labelling function*. This function is defined as $\lambda : T \to \Sigma \cup \{\epsilon\}$, i.e. $\lambda(t) = e$ or $\lambda(t) = \epsilon$ for $t \in T$. Also, this labelling function can be extended to the Kleene closure of Σ by $\lambda : T^* \to \Sigma^*$ where for each sequence of transitions σ and transition t, $\lambda(\sigma t) = \lambda(\sigma)\lambda(t)$. A *labelled* Petri net is defined as a four tuple $(\mathcal{N}, M_0, \Sigma, \lambda)$ in which we associate to each label $e \in \Sigma$ a set of transitions $\tau(e)$.

$$\tau(e) = \{t \mid t \in T, e = \lambda(t)\} \tag{2}$$

2.2. Integer Fourier–Motzkin elimination method

The elimination of a variable from a set of inequalities $I := A\mathbf{x} \leq \mathbf{b}$, where $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ can be achieved by Fourier–Motzkin elimination (FME) method (Dantzig, 1972; Duffin, 1974). The variables are eliminated one by one as explained as follows. It is sufficient to describe the process of eliminating one variable, as the same procedure can be repeatedly applied to eliminate the required number of variables. Also, for the sake of simplicity, all entries in the last column of A are assumed to be 0, $+1$ or -1. Assuming that x_n is to be eliminated, I can be rewritten as shown in (3):

$$\begin{aligned} I^0 &: \quad \mathbf{a}'_{\mathbf{i}}\mathbf{x}' \leq b_i,\ i = 1, \dots, m_1 \\ I^- &: \quad \mathbf{a}'_{\mathbf{j}}\mathbf{x}' - x_n \leq b_j,\ j = m_1 + 1, \dots, m_2 \\ I^+ &: \quad \mathbf{a}'_{\mathbf{k}}\mathbf{x}' + x_n \leq b_k,\ k = m_2 + 1, \dots, m \end{aligned} \tag{3}$$

where $\mathbf{x}' = \{x_1, x_2, \ldots x_{n-1}\}$, i.e. the same set of variables without x_n. Also I^0, I^- and I^+ are sets of inequalities in I which have zero, negative and positive coefficients of x_n. If I^+ is empty, then all inequalities in I^- can simply be deleted. Likewise, if I^- is empty, then all inequalities in I^+ can be discarded. Assume that $l = \max(\mathbf{a}'_j\mathbf{x}' - b_j, j = m_1+1, \ldots, m_2)$ and $u = \min(b_k - \mathbf{a}'_k\mathbf{x}', k = m_2+1, \ldots, m)$. Since the last two lines of (3) are equivalent to $l \le x_n \le u$, the variable x_n can be eliminated. This yields the *reduced* set R in (4) with no x_n as an equivalent to (3):

$$\begin{aligned} \mathbf{a}'_i\mathbf{x}' &\le b_i, \ \ i = 1, \ldots, m_1 \\ \mathbf{a}'_j\mathbf{x}' - b_j &\le b_k - \mathbf{a}'_k\mathbf{x}', \ \ j = m_1+1, \ldots, m_2, \\ k &= m_2+1, \ldots, m \end{aligned} \tag{4}$$

Theorem 1 (*Duffin, 1974*)**.** *Assume that the variables $x_{k+1}, \ldots, x_n$ have been eliminated in order by using the FME method described above from a set of linear inequalities I. This results in the reduced set R. Then $\alpha_1, \ldots, \alpha_k$ is a solution of R iff there exist values $\alpha_{k+1}, \ldots, \alpha_n$ such that $\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n$ is a solution of I.*

This theorem represents an important result for the purpose of fault diagnosis, as will be clear in the following sections. An extension of this result to a set of inequalities having integer-valued variables has been reported in Pugh (1991) and Williams (1976). This extension, named Integer FME (IFME), is to ensure that for any integer solution in R, there exists an integer solution in I. In this paper, we have chosen the method presented in Pugh (1991), which better meets our need as it is somewhat simpler and more efficient.

3. Problem statement

In this section, a description of the problem of fault diagnosis in DES modelled by labelled Petri nets is given based on the formulation adopted by Cabasino et al. (2011) and Fanti et al. (2013). Consider a labelled Petri net $(\mathcal{N}, M_0, \Sigma, \lambda)$, as defined in Section 2.1. Suppose that the set of transitions T in $\mathcal{N}$ is partitioned into two sets: observable transitions T_o and unobservable transitions T_u. We further assume that faults are unobservable transitions, i.e. $T_f \subseteq T_u$, in which T_f is the set of transitions which are modelling occurrences of faults. The set T_u may have other transitions which model no fault, i.e. they model normal events.

Consider also the *projection* function $\pi : T \to T_o \cup \{\epsilon\}$ that maps unobservable transitions to the empty string ϵ, i.e. $\pi(t) = \epsilon$ for $t \in T_u$, while $\pi(t) = t$ for $t \in T_o$. The projection function π can be extended to the Kleene-closure of T by $\pi : T^* \to (T_o \cup \{\epsilon\})^*$, where for each sequence of transitions $\sigma \in T^*$ and each transition t, $\pi(\sigma t) = \pi(\sigma)\pi(t)$. We assume $\pi(\epsilon) = \epsilon$ and that $\pi(t\epsilon) = \pi(\epsilon t) = \epsilon$ for each $t \in T_u$. Moreover, the *inverse projection* function is defined as $\pi^{-1} : T_o^* \to 2^{\{\sigma \in L(\mathcal{N}, M_0) | \pi(\sigma) = \mathbf{s}, \ \mathbf{s} \in T_o^*\}}$. A *legal sequence* $\mathbf{s} \in T_o^*$ is such that $\pi^{-1}(\mathbf{s}) \neq \emptyset$.

Let $\omega \in \Sigma^*$ denote an observed sequence of events (labels), where $\omega = \lambda(\mathbf{s})$ and $\mathbf{s} = \pi(\sigma)$ for a given sequence $\sigma \in T^*$. To simplify the presentation of this paper, we only consider one type of fault $T_f = \{t_1, t_2, \ldots, t_k\}$; the extension to multiple types is straightforward. In particular, to create a set of inequalities for a given fault type, the transitions representing faults in the other fault types are considered as normal unobservable transitions. Since it is not required to uniquely identify occurrences of every fault of T_f, a firing of any transition $t \in T_f$ implies that a fault has occurred. We suppose that the labels captured by ω are the only information we receive when a sequence of observable transitions fires. A *diagnoser* (as formally defined in the following sections) uses such information to identify if a fault has occurred or may have occurred.

In this paper, the problem of fault diagnosis is addressed with the assumption that different transitions could share the same label, taking into account that these transitions might be simultaneously enabled.

4. The IFME method for fault diagnosis in LPN

The main results obtained in this paper are covered in this section. In order to formulate the IFME-based solution, we first introduce some of necessary definitions and notation.

4.1. Definitions and notations

The IFME-based approach for fault diagnosis essentially relies on using inequalities. The enabling conditions of Petri nets can be formed as a set of inequalities. Besides, the presence and absence of faults can be expressed in the form of inequalities. Suppose that transition $t_i \in T$ is a fault transition. Then t_i does not appear in a firing sequence σ if and only if $\mathbf{c} := \#(t_i, \sigma) = 0$ holds. Also, the occurrence of t_i in σ can be trivially written as $\neg\mathbf{c} := \#(t_i, \sigma) > 0$, i.e., the negation of $\mathbf{c}$. In addition, we can represent a set of faults as inequalities by extending the formulation above. Recall that $T_f = \{t_1, t_2, \ldots, t_r\}$ is a fault type; we associate two inequalities $\neg\mathbf{c} := \sum_{t \in T_f} \#(t, \sigma) > 0$ and $\mathbf{c} := \sum_{t \in T_f} \#(t, \sigma) \le 0$. Then, no fault of T_f appearing in σ implies that $\mathbf{c}$ holds. In contrast, a fault of T_f appears in σ implies that $\neg\mathbf{c}$ holds. Next, two definitions are introduced for use in determining the set $X(\omega)$ described below.

Definition 2. Suppose that $\mathbf{e}$ is an inequality of the form $a_1x_1 + \cdots + a_nx_n \le b$ in the variables set $\mathbf{x} = (x_1, \ldots, x_n), x_i \in \mathbb{N}$ and $a_1, \ldots, a_n, b \in \mathbb{Z}$. Consider the values $\alpha_1, \ldots, \alpha_n$ assigned to $x_1, \ldots, x_n$, respectively. Supposing that $\nu = (\alpha_1, \ldots, \alpha_n)$, then the notation $\nu \vDash \mathbf{e}$ means that ν satisfies the inequality $\mathbf{e}$ if and only if $a_1\alpha_1 + \cdots + a_n\alpha_n \le b$ is true.

Definition 3. *The diagnosis labelling function*: a diagnosis labelling function $D : T_o^* \times 2^{T_f} \to \{N, F, FN\}$ is a mapping that associates to each sequence of observable transitions $\mathbf{s}$ with respect to the fault type T_f (expressed by $\mathbf{c}$), one of the following diagnosis labels:

- $D(\mathbf{s}, T_f) = N$ if $\forall\sigma \in L(\mathcal{N}, M_0)$ such that $\pi(\sigma) = \mathbf{s}$, $\#(\sigma) \vDash \mathbf{c}$ holds.
- $D(\mathbf{s}, T_f) = F$ if $\forall\sigma \in L(\mathcal{N}, M_0)$ such that $\pi(\sigma) = \mathbf{s}$, $\#(\sigma) \vDash \neg\mathbf{c}$ holds.
- $D(\mathbf{s}, T_f) = FN$ if there exist two sequences $\sigma_1, \sigma_2 \in L(\mathcal{N}, M_0)$ such that $\pi(\sigma_1) = \pi(\sigma_2) = \mathbf{s}$, but $\#(\sigma_1) \vDash \mathbf{c}$ and $\#(\sigma_2) \vDash \neg\mathbf{c}$ hold.

Two sets of sequences are defined in the following. The first set characterises the set of sequences in the language of $\mathcal{N}$ corresponding to an observed sequence of events ω as shown below:

$$\Gamma(\omega) = \{\sigma \in L(\mathcal{N}, M_0) \ | \mathbf{s} = \pi(\sigma), \omega = \lambda(\mathbf{s})\} \tag{5}$$

The second set consists of a number of pairs associated with a given sequence of observed events. Each pair captures the form (*observed sequence*, *diagnosis label*) expressed in the following definition:

Definition 4. Suppose that $(\mathcal{N}, M_0, \Sigma, \lambda)$ is a labelled Petri net. Given an observed sequence $\omega \in \Sigma^*$, we define a set of pairs associated with ω with respect to the fault type T_f as:

$$X(\omega) = \{(\mathbf{s}, l) \ | \exists\sigma \in \Gamma(\omega), \mathbf{s} = \pi(\sigma), l = D(\mathbf{s}, T_f)\} \tag{6}$$

Note that the set $X(\omega) \neq \emptyset$ because ω corresponds to a firing sequence. In the following, the definition of diagnoser is extended inspired by definitions presented in Cabasino et al. (2011) and Fanti et al. (2013).

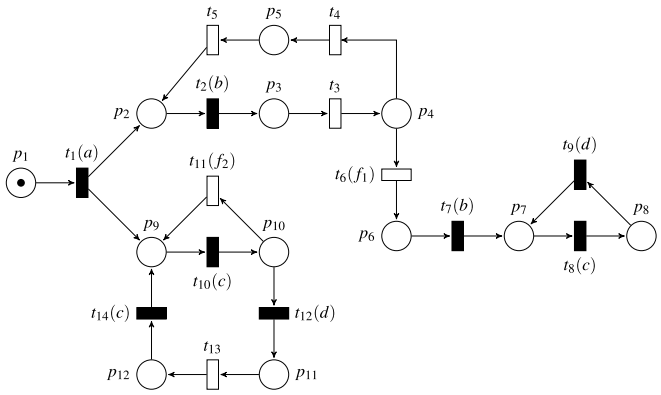

Fig. 1. A labelled Petri net example.

Definition 5. A diagnoser is a function $\Delta : \Sigma^* \times 2^{T_f} \to \{NoFault, Faulty, Uncertain\}$ that associates with each observed sequence $\omega \in \Sigma^*$ with respect to the fault type T_f one of the following diagnosis states:

- $\Delta(\omega, T_f) = NoFault$ if $\forall \sigma \in \Gamma(\omega)$, $\#(\sigma) \models \mathbf{c}$ holds. This state indicates that there is no sequence having the same labels as ω containing a fault transition in T_f, i.e. no fault has occurred.
- $\Delta(\omega, T_f) = Faulty$ if $\forall \sigma \in \Gamma(\omega)$, $\#(\sigma) \models \neg\mathbf{c}$ holds. This state is *Faulty* as all sequences having the same labels as ω contain a fault transition in T_f.
- $\Delta(\omega, T_f) = Uncertain$ if there exists two sequences $\sigma_1, \sigma_2 \in \Gamma(\omega)$ such that $\#(\sigma_1) \models \mathbf{c}$ and $\#(\sigma_2) \models \neg\mathbf{c}$ hold. In this case, the behaviour of the system is ambiguous because both *NoFault* and *Faulty* states are possible during the observed sequence.

Example 1. Consider the labelled Petri net depicted in Fig. 1. In this net, the initial marking is $M_0 = [100000000000]$. In the figure, the set of observable transitions is depicted by solid rectangles, while empty rectangles represent unobservable transitions. The labelling function λ yields $\tau(\epsilon) = \{t_3, t_4, t_5, t_6, t_{11}, t_{13}\}$, $\tau(a) = \{t_1\}$, $\tau(b) = \{t_2, t_7\}$, $\tau(c) = \{t_8, t_{10}, t_{14}\}$ and $\tau(d) = \{t_9, t_{12}\}$. Moreover, there is one fault type having two fault transitions t_6 and t_{11} denoted by f_1 and f_2, respectively as shown in the figure. Thus, we have one constraint $\mathbf{c} := x_6 + x_{11} \leq 0$ and its negation $\neg\mathbf{c} := x_6 + x_{11} > 0$ (also written as $\neg\mathbf{c} := -x_6 - x_{11} \leq -1$). Note that in this Petri net, two transitions sharing the same label could be enabled simultaneously, e.g. the transitions t_8 and t_{10}.

If we suppose that $\omega = a$, then $\Gamma(\omega) = \{t_1\}$. In which case, we are certain that no fault from T_f has occurred, i.e. $\Delta(a, T_f) = NoFault$. Assuming now that $\omega = abb$, then $\Gamma(\omega) = \{t_1t_2t_3t_4t_5t_2, t_1t_2t_3t_6t_7\}$. One of these sequences has the fault transition t_6, but the others have none. Hence, $\Delta(abb, T_f) = Uncertain$. When observing $\omega = acc$, a different diagnosis state is obtained. In effect, $\Gamma(\omega) = \{t_1t_{10}t_{11}t_{10}\}$. This ensures that a fault (t_{11}) from T_f has occurred. Formally, $\Delta(acc, T_f) = Faulty$.

We end this section by recalling the results obtained in Dotoli et al. (2009) in the case of Petri nets as expressed in the following proposition.

Proposition 1 (*Dotoli et al., 2009*). *Given a Petri net $(\mathcal{N}, M_0)$ having no cycle of unobservable transitions and an observed sequence of transitions $\mathbf{s} \in T_o^*$. Then, there exists a sequence $\sigma = \sigma_1 t_1 \dots \sigma_h t_h$ such that $M_0 \xrightarrow{\sigma_1 t_1} M_1 \to \cdots \to M_{h-1} \xrightarrow{\sigma_h t_h} M_h$ and $\mathbf{s} = t_1 \dots t_h$ for $\sigma_1, \dots, \sigma_h \in T_u^*$ if and only if there exists a solution $\#(\sigma_1), \dots, \#(\sigma_h)$ to the following set of inequalities:*

$$\mathscr{S} = \begin{cases} A_u \cdot \#(\sigma_1) \geq Pre(., t_1) - M_0 & (1) \\ A_u \cdot (\#(\sigma_1) + \#(\sigma_2)) \geq Pre(., t_2) - M_0 - A \cdot \mathbf{u}_1 & (2) \\ \quad \vdots & \\ A_u \sum\limits_{1 \leq i \leq h} \#(\sigma_i) \geq Pre(., t_h) - M_0 - A \sum\limits_{1 \leq i \leq h-1} \mathbf{u}_i & (h) \end{cases}$$

where A_u is the restriction of A on the unobservable transitions and $\mathbf{u}_i$ is the firing vector of t_i for $i = 1, \dots, h-1$.

From Proposition 1, we can infer that if the set of inequalities $\mathscr{S}$ does not have a solution with respect to $\mathbf{s} = t_1 \dots t_h$, then there does not exist a corresponding sequence $\sigma \in L(\mathcal{N}, M_0)$ such that $\sigma = \sigma_1 t_1 \dots \sigma_h t_h$. The set of inequalities in $\mathscr{S}$ can also be rewritten as:

$$\mathscr{S}' = \begin{cases} -A_u \cdot \#(\sigma_1) + Pre(., t_1) \leq M_0 & (1) \\ -A_u \cdot (\#(\sigma_1) + \#(\sigma_2)) - A \cdot \mathbf{u}_1 + Pre(., t_2) \leq M_0 & (2) \\ \quad \vdots & \\ -A_u \sum\limits_{1 \leq i \leq h} \#(\sigma_i) - A \sum\limits_{1 \leq i \leq h-1} \mathbf{u}_i + Pre(., t_h) \leq M_0 & (h) \end{cases}$$

where each subset $\mathscr{S}'_i$, $i = 1, \dots, h$, of inequalities in $\mathscr{S}'$, e.g. $\mathscr{S}'_1 = -A_u \cdot \#(\sigma_1) + Pre(., t_1) \leq M_0$, can simply be represented by the following general form:

$$I := (-A \cdot \mathbf{x}) + \mathbf{y} \leq M_0 \tag{7}$$

given a sequence of transitions $\sigma_1 t_1 \dots \sigma_i t_i$, where $\mathbf{y} = Pre(., t_i)$ and $\mathbf{x} = \#(\sigma_1 t_1 \dots \sigma_i)$. If we assume that the sequence $\sigma_1 t_1 \dots \sigma_i$ is enabled at M_0, then the transition t_i is enabled if (7) holds.

4.2. Identification of the legal sequences

Given the set of inequalities I as defined in Section 4.1 in the sets of variables $\mathbf{x}$ and $\mathbf{y}$. Then, assume that the *IFME* is applied to I to eliminate the variables corresponding to the unobservable transitions resulting in the set of inequalities I'. We present the following proposition to characterise legal sequences (sequences of observable transitions). In other words, this proposition can be applied to decide whether a sequence of observable transitions has at least one corresponding sequence in a labelled Petri net.

Proposition 2. *Suppose that $(\mathcal{N}, M_0, \Sigma, \lambda)$ is a labelled Petri net having no cycle of unobservable transitions. Also, assume that I is the set of inequalities of (7) in the sets of variables $\mathbf{x}$ and $\mathbf{y}$. The set of inequalities I' is as defined above. Then, for any given sequence of observable transitions $\mathbf{s} = t_1 \dots t_h$, there exists a corresponding sequence $\sigma = \sigma_1 t_1 \dots \sigma_h t_h$ in $\mathcal{N}$ such that $M_0 \xrightarrow{\sigma_1 t_1} M_1 \to \cdots \xrightarrow{\sigma_h t_h} M_h$ iff there exists a vector $\nu' = (\alpha_1, \dots, \alpha_k, Pre(p_1, t), \dots, Pre(p_m, t)) \models I'$, where $\alpha_i = \#(t_i, \mathbf{s}')$, $\mathbf{s}' = t_1 \dots t_{h-1}$ and $k = |T_o|$.*

Proof. *Necessity*: If there exists σ such that $\pi(\sigma) = \mathbf{s}$, then there exists $\nu = \#(\sigma)$ such that $\nu \models I$ by the enabling condition. As a result, there exists a corresponding ν' such that $\nu' \models I'$ by Theorem 1.

Sufficiency: If there exists $\nu' \models I'$, then there exists a corresponding sequence in $\mathcal{N}$. We prove this case by the induction on the length of $\mathbf{s}$, denoted by $|\mathbf{s}|$ as follows:

Base case: Assume that $|\mathbf{s}| = 1$. If $(\alpha_1, \dots, \alpha_k, Pre(p_1, t_1), \dots, Pre(p_m, t_1)) \models I'$, where $\alpha_i = 0$ for $1 \leq i \leq k$, then

there exists a solution $\nu = (\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n, Pre(p_1, t_1), \ldots, Pre(p_m, t_1)) \vDash I$ by Theorem 1. Assume that $\nu = (\alpha_1, \ldots, \alpha_n)$, then the subnet $\mathcal{N}_\nu$ has only unobservable transitions. Since $\mathcal{N}_\nu$ is cycle free by the assumption, there exists a sequence $\sigma_1 \in T_u^*$ such that $M_0 \xrightarrow{\sigma_1} M$ and $\#(\sigma_1) = \nu$ by Lemma 1. As a result, we have a sequence $\sigma_1 t_1$ such that $M_0 \xrightarrow{\sigma_1 t_1} M_1$ for $\mathbf{s} = t_1$. This proves the case.

Induction step: Suppose that the result holds for all $\mathbf{s}$ with $|\mathbf{s}| < h$ (Induction hypothesis). Then, we prove that the result holds for $|\mathbf{s}| = h$. Hence, for $\mathbf{s}' = t_1 \ldots t_{h-1}$ there exists a sequence $\sigma' = \sigma_1 t_1 \ldots \sigma_{h-1} t_{h-1}$ such that $M_0 \xrightarrow{\sigma_1 t_1} M_1 \rightarrow \cdots \xrightarrow{\sigma_{h-1} t_{h-1}} M_{h-1}$. If we have $\mathbf{s} = \mathbf{s}' t_h$ such that $(\alpha_1, \ldots, \alpha_k, Pre(p_1, t_h), \ldots, Pre(p_m, t_h)) \vDash I'$, then there exists a solution $\nu = (\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n, Pre(p_1, t_h), \ldots, Pre(p_m, t_h)) \vDash I$ by Theorem 1. Assume that $\nu' = (\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n)$ and $\mathbf{z} = \nu' - \#(\sigma')$, $\mathbf{z} \in \mathbb{N}^n$, then $M = M_{h-1} + A\mathbf{z} \geq \mathbf{0}$. Since the subnet $\mathcal{N}_\mathbf{z}$ has only unobservable transitions and it is cycle free, there exists a sequence σ_h such that $M_{h-1} \xrightarrow{\sigma_h} M$ with $\#(\sigma_h) = \mathbf{z}$. Further, since $\nu = (\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n, Pre(p_1, t_h), \ldots, Pre(p_m, t_h)) \vDash I$, then $M \xrightarrow{t_h} M_h$. Consequently, there exists a sequence $\sigma = \sigma_1 t_1 \ldots \sigma_h t_h$ in $\mathcal{N}$ such that $\mathbf{s} = t_1 \ldots t_h$. This also proves this case. □

Proposition 2 gives a complete procedure to identify the legal sequences. Identification of these sequences is necessary to determine the diagnosis states.

4.3. Computing the diagnosis states

Suppose that the set of fault transitions in $\mathcal{N}$ is $T_f \subseteq T_u$ and all faults are of the same type. We can further suppose that I is as defined in (7) in variables $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{c}$ and $\neg\mathbf{c}$ as defined in Section 4.1. In order to compute the diagnosis state, we first create two sets $I \cup \{\mathbf{c}\}$ and $I \cup \{\neg\mathbf{c}\}$. Then, applying the IFME method to the sets $I \cup \{\mathbf{c}\}$ and $I \cup \{\neg\mathbf{c}\}$ respectively yields the reduced sets R and R' created by eliminating every variable corresponding to a transition in the set T_u. In the following, we present the results that capture the details of computing a diagnosis state upon observing a sequence of events ω.

Theorem 2. *Suppose that $(\mathcal{N}, M_0, \Sigma, \lambda)$ is a labelled Petri net having no cycle of unobservable transitions. Also, assume that the set of inequalities I is as defined in (7). The sets of inequalities R and R' plus the inequalities $\mathbf{c}$ and $\neg\mathbf{c}$ are described above. Then, for any given sequence of observable transitions $\mathbf{s} = \mathbf{s}'t = \pi(\sigma)$ and $t \in T_o$ such that there exists $\sigma \in L(\mathcal{N}, M_0)$, consider that $\nu' = (\alpha_1, \ldots, \alpha_k, Pre(p_1, t), \ldots, Pre(p_m, t))$ is a vector, where $\alpha_i = \#(t_i, \mathbf{s}')$, $\mathbf{s}' = t_1 \ldots t_{h-1}$ and $k = |T_o|$. Then $D(\mathbf{s}, T_f)$ is determined as follows:*

$$D(\mathbf{s}, T_f) = \begin{cases} N & \text{iff } \nu' \nvDash R' \\ F & \text{iff } \nu' \nvDash R \\ FN & \text{iff } \nu' \vDash R \wedge \nu' \vDash R' \\ Impossible & \text{iff } \nu' \nvDash R \wedge \nu' \nvDash R' \end{cases}$$

Proof. Case (i) $D(\mathbf{s}, T_f) = N$: By contradiction, assume that $\nu' \nvDash R'$, but $D(\mathbf{s}, T_f)$ is not N. If $\nu' \nvDash R'$, then there does not exist a corresponding solution of ν' in $I \cup \{\neg\mathbf{c}\}$ by Theorem 1. But ν' has a corresponding solution, say ν, in I because it is coming from a sequence in $L(\mathcal{N}, M_0)$, see Section 2.1. Thus, $\nu \nvDash \neg\mathbf{c}$, i.e. $\nu \vDash \mathbf{c}$. As a result, $\forall\sigma' \in L(\mathcal{N}, M_0)$ such that $\pi(\sigma') = \mathbf{s}$, $\#(\sigma') \vDash \mathbf{c}_i$ holds. Hence $D(\mathbf{s}, T_f)$ is N, see Definition 3. This contradicts the assumption. The converse is also true.

Case (ii) $D(\mathbf{s}, T_f) = F$: Using a similar argument in the proof of Case i by replacing R' with R, we can prove this case.

Algorithm 1 : build the diagnoser (offline step).

Input: A labelled Petri net $(\mathcal{N}, M_0, \Sigma, \lambda)$, a set of unobservable transitions T_u, a single fault type T_f.
Output: The pair (R, R') plus the set I'.

1: **Let** $I \leftarrow -A\mathbf{x} + Pre(., t) \leq M_0$
2: **Let** $\mathbf{c} \leftarrow \sum_{t_j \in T_f} x_j \leq 0$, $\neg\mathbf{c} \leftarrow \sum_{t_j \in T_f} -x_j \leq -1$
3: $I' \leftarrow I$
4: $R \leftarrow I \cup \{\mathbf{c}\}$
5: $R' \leftarrow I \cup \{\neg\mathbf{c}\}$
6: **for all** $t_j \in T_u$ **do**
7: $I' \leftarrow IFME\_method(I', x_j)$
8: $R \leftarrow IFME\_method(R, x_j)$
9: $R' \leftarrow IFME\_method(R', x_j)$
10: **end for**

Case (iii) $D(\mathbf{s}, T_f) = FN$: If $\nu' \vDash R$, then there exists a corresponding solution in $\nu \vDash I \cup \{\mathbf{c}\}$ by Theorem 1. Hence, there exists a sequence in $L(\mathcal{N}, M_0)$ which satisfies $\mathbf{c}$. Likewise, we can prove that if $\nu' \vDash R'$, then there exists another sequence satisfying $\neg\mathbf{c}$. Since there are two sequences having the same $\mathbf{s}$, but one of them satisfies $\mathbf{c}$ and the other satisfies $\neg\mathbf{c}$, we have $D(\mathbf{s}, T_f) = FN$, see Definition 3. The converse is also true.

Case (iv) *Impossible*: It is a contradictory statement to have ν', which corresponds to an observed sequence, that does not satisfy $\mathbf{c}$ and $\neg\mathbf{c}$ at the same time. The converse is also true and this completes the proof. □

Corollary 1. *Assume that $(\mathcal{N}, M_0, \Sigma, \lambda)$ is a labelled Petri net. Then, for any given sequence of observed events $\omega \in \Sigma^*$, considering that the set $X(\omega)$ is such that each $(\mathbf{s}, l) \in X(\omega)$ is legal, $\Delta(\omega, T_f)$ is determined as follows:*

$$\Delta(\omega, T_f) = \begin{cases} NoFault & \text{iff } \forall(\mathbf{s}, l) \in X(\omega), l = N \\ Faulty & \text{iff } \forall(\mathbf{s}, l) \in X(\omega), l = F \\ Uncertain & Otherwise \end{cases}$$

Proof. A direct proof. □

4.4. Fault diagnosis algorithms

In this section, the algorithms developed for fault diagnosis in labelled Petri nets are described. In Algorithm 1, steps 7–9 recursively invoke the IFME procedure (explained previously in Section 2.2) with two parameters. The first parameter represents the set of inequalities and the second one is the variable to be eliminated from this set. The output of Algorithm 1 consists of sets of inequalities I', R and R'.

The input of Algorithm 2 is the fault type T_f and $\tau(e)\forall e \in \Sigma$, in addition to sets of inequalities I', R and R'. The output of the algorithm is a diagnosis state from $\{NoFault, Faulty, Uncertain\}$ (see Definition 5). This algorithm starts by initialising ω' and $X(\omega')$. Then, in step 2 in particular, the algorithm enters into a loop to estimate the diagnosis state. In step 3, the algorithm waits until a new event e is observed and then adds it to the previous sequence ω', creating the sequence ω. From step 5 to step 21, the algorithm builds the set $X(\omega)$. First, the set of all sequences $\mathbf{s} \in T_o^*$ corresponding to ω in $\mathcal{N}$ is generated in steps 6–8. The variables $x_1, \ldots, x_k, y_1, \ldots, y_m$ are computed and their values are allocated to the vector ν' (step 9). Then, each generated sequence is checked to determine whether it has a corresponding sequence in the Petri net (step 10), see Proposition 2. The function $D(\mathbf{s}, T_f)$ is computed in steps 11–17 by applying Theorem 2. Steps 22–28 determine the diagnosis state $\Delta(\omega, T_f)$ based on Corollary 1.

Algorithm 2 : fault diagnosis (online step).

Input: A single fault type T_f; $\tau(e), \forall e \in \Sigma$
and the sets R, R' and I' as defined in Algorithm 1.
Output: A diagnosis state {*NoFault*, *Faulty*, *Uncertain*}.

Initialise $\omega' = \epsilon$, $X(\omega') = \emptyset$
loop
  if a new event e is observed **then**
    Let $\omega \leftarrow \omega' e$
    Initialise $X(\omega) \leftarrow \emptyset$
    for all $t \in \tau(e)$ **do**
      for all $\mathbf{s}' \in X(\omega')$ **do**
        $\mathbf{s} \leftarrow \mathbf{s}'t$
        $\nu' \leftarrow (\#(\mathbf{s}'), Pre(p_1, t), \ldots, Pre(p_m, t))$
        if $\nu' \vDash I'$ **then**
          if $\nu' \nvDash R'$ **then**
            $D(\mathbf{s}, T_f) \leftarrow N$
          else if $\nu' \nvDash R$ **then**
            $D(\mathbf{s}, T_f) \leftarrow F$
          else if $\nu' \vDash R$ **and** $\nu' \vDash R'$ **then**
            $D(\mathbf{s}, T_f) \leftarrow FN$
          end if
          $X(\omega) \leftarrow X(\omega) \cup \{(\mathbf{s}, D(\mathbf{s}, T_f))\}$
        end if
      end for
    end for
    if $\forall(\mathbf{s}, l) \in X(\omega)$, $l = N$ **then**
      $\Delta(\omega, T_f) \leftarrow NoFault$
    else if $\forall(\mathbf{s}, l) \in X(\omega)$, $l = F$ **then**
      $\Delta(\omega, T_f) \leftarrow Faulty$
    else
      $\Delta(\omega, T_f) \leftarrow Uncertain$
    end if
  end if
  $\omega' \leftarrow \omega$, $X(\omega') \leftarrow X(\omega)$
end loop

Table 1
The sets of inequalities I and I' of the net in Fig. 1.

I	$I' \leftarrow IFME(I)$
$x_1 + y_1 \le 1$	$x_1 + y_1 \le 1$
$-x_1 + x_2 - x_5 + y_2 \le 0$	$-x_2 + y_3 \le 0$
$-x_2 + x_3 + y_3 \le 0$	$-x_{12} + y_{11} \le 0$
$-x_3 + x_4 + x_6 + y_4 \le 0$	$-x_8 + x_9 + y_8 \le 0$
$-x_4 + x_5 + y_5 \le 0$	$-x_2 + y_3 + y_4 \le 0$
$-x_6 + x_7 + y_6 \le 0$	$-x_{10} + x_{12} + y_{10} \le 0$
$-x_7 + x_8 - x_9 + y_7 \le 0$	$-x_7 + x_8 - x_9 + y_7 \le 0$
$-x_8 + x_9 + y_8 \le 0$	$-x_2 + y_3 + y_4 + y_5 \le 0$
$-x_1 + x_{10} - x_{11} - x_{14} + y_9 \le 0$	$-x_{12} + x_{14} + y_{11} + y_{12} \le 0$
$-x_{10} + x_{11} + x_{12} + y_{10} \le 0$	$-x_2 + x_7 + y_3 + y_4 + y_6 \le 0$
$-x_{12} + x_{13} + y_{11} \le 0$	$-x_1 + y_2 + y_3 + y_4 + y_5 \le 0$
$-x_{13} + x_{14} + y_{12} \le 0$	$-x_1 + x_{12} - x_{14} + y_9 + y_{10} \le 0$
$-x_i \le 0 \mid_{i \in \{3,4,5,6,11,13\}}$	$-x_2 + x_7 + y_3 + y_4 + y_5 + y_6 \le 0$
	$-x_1 + x_7 + y_2 + y_3 + y_4 + y_5 + y_6 \le 0$

4.5. Computational complexity

Using IFME to produce such a diagnoser (Algorithm 1), the number of inequalities may grow in each elimination step. For instance, the set of inequalities after the first elimination could have $(\frac{m}{2})^2$ in the worst case, where m is the number of inequalities in the initial set. The final set of inequalities after eliminating k_1 variables (where k_1 is the number of unobservable transitions) could have $O(m^{2^{k_1}})$ in the worst case.

Let us consider the computational complexity to compute the diagnosis (Algorithm 2). This complexity relies on the number of observed events and the size of the diagnoser. To be precise, assume that m_F is the number of inequalities in $I' \cup R \cup R'$ of the fault type T_f, then the online step requires in the worst case $O(|X(\omega')| \cdot |\tau(e)| \cdot m_F)$ to decide the diagnosis state. Note that $|X(\omega')| \le |T_o| \cdot n_1$, where n_1 is the length of the sequence ω'.

We provide a brief comparison in terms of the computational complexity between the IFME-based approach and the ILP-based approaches. The latter requires solving a set of ILP problems online, each of which costs an exponential time in the number of observed events. While the IFME-based approach requires a number of verification processes against a set of inequalities, in each verification, we only require polynomial time in the number of observed events.

4.6. Illustrative example

Recalling the labelled Petri net of Fig. 1, three sets of inequalities are to be created to represent the diagnoser. We start by extending the set of inequalities I by adding the inequalities $\mathbf{c} := x_6 + x_{11} \le 0$ and $\neg\mathbf{c} := -x_6 - x_{11} \le -1$ in order to obtain $I \cup \{\mathbf{c}\}$ and $I \cup \{\neg\mathbf{c}\}$, respectively. Applying the IFME method to the three sets I, $I \cup \{\mathbf{c}\}$ and $I \cup \{\neg\mathbf{c}\}$ results in the sets I', R and R' as shown in Tables 1 and 2. The resulting sets of inequalities are in the set of variables $\{x_1, x_2, x_7, x_8, x_9, x_{10}, x_{12}, x_{14}\}$ plus the set of variables $\{y_j \mid 1 \le j \le 12\}$.

Now, suppose that we observe the sequence $\omega = ab$. Two potential sequences $\mathbf{s}_1 = t_1 t_2$ and $\mathbf{s}_2 = t_1 t_7$ could correspond ab. The vector ν' can be computed for $\mathbf{s}_1$ and $\mathbf{s}_2$ as follows. Assume that $\mathbf{s}_1 = \mathbf{s}'_1 t_2$ and $\mathbf{s}_2 = \mathbf{s}'_2 t_7$. In case of $\mathbf{s}_1$, we obtain $\#(t_1, \mathbf{s}'_1) = 1$ and $\#(t_i, \mathbf{s}'_1) = 0$, $\forall t_i \in \{2, 7, 8, 9, 10, 12, 14\}$; also $Pre(p_1, t_2) = 1$ and $Pre(p_j, t_2) = 0$, $\forall j = 2, \ldots, 12$. For the sequence $\mathbf{s}_2$, we obtain $\#(t_1, \mathbf{s}'_2) = 1$ and $\#(t_i, \mathbf{s}'_2) = 0$, $\forall t_i \in \{2, 7, 8, 9, 10, 12, 14\}$; also $Pre(p_6, t_7) = 1$ and $Pre(p_j, t_7) = 0$, $\forall j = \{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12\}$. Hence, the vectors $\nu'_1 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ and $\nu'_2 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$ are determined for $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. Since $\nu'_1 \vDash I'$, $\mathbf{s}_1$ is a legal sequence, but $\mathbf{s}_2$ is not (see Proposition 2). Thus, we ignore $\mathbf{s}_2$ and check ν'_1 against R and R'; we find that $\nu'_1 \vDash R$ and $\nu'_1 \nvDash R'$. This implies that $D(\mathbf{s}_1, T_f) = N$ (see Theorem 2). Based on this, the set $X(ab) = \{(t_1 t_2, N)\}$. Since $X(ab)$ contains one sequence with diagnosis label N, we have *NoFault* diagnosis state (see Corollary 1).

5. Conclusion

We have presented a new approach for fault diagnosis under partial observation in labelled Petri net models of DES. This approach adopts the IFME method to build the diagnoser offline. In particular, this paper addresses the most general case of fault diagnosis in Petri nets in which another source of non-determinism originates from the fact that different transitions could share the same label and these transitions could be indistinguishable. As a result, part of computational effort is required online to handle this case. By observing a sequence of events (labels), a set of sequences of transitions corresponding to these observed sequences is generated. Then, using the diagnoser this set is analysed to make diagnosis decisions. Since the diagnoser is no longer represented as an automaton, the IFME-based approach can be used in both finite and infinite systems. Furthermore, this current representation of the diagnoser makes the computational complexity of our approach heavily rely on the number of unobservable transitions and not state space size.

A future direction of research can investigate the diagnosis of more complex forms and other types of faults. In addition, decentralised and distributed diagnosis, where many local diagnosers could monitor the state of the system will be taken into account.

Table 2
The sets of inequalities R and R' of the net in Fig. 1.

$R \leftarrow IFME(I \cup \{\mathbf{c}\})$	$R' \leftarrow IFME(I \cup \{\neg\mathbf{c}\})$
$x_1 + y_1 \leq 1$	$x_1 + y_1 \leq 1$
$-x_7 + x_8 - x_9 + y_7 \leq 0$	$-x_7 + x_8 - x_9 + y_7 \leq 0$
$-x_8 + x_9 + y_8 \leq 0$	$-x_8 + x_9 + y_8 \leq 0$
$-x_2 + y_3 \leq 0$	$-x_2 + y_3 \leq 0$
$-x_2 + x_7 + y_3 + y_4 + y_6 \leq 0$	$-x_2 + x_7 + y_3 + y_4 + y_6 \leq 0$
$-x_2 + y_3 + y_4 \leq 0$	$-x_2 + y_3 + y_4 \leq 0$
$-x_1 + x_7 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 0$	$-x_1 + x_7 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 0$
$-x_1 + y_2 + y_3 + y_4 + y_5 \leq 0$	$-x_1 + y_2 + y_3 + y_4 + y_5 \leq 0$
$-x_2 + x_7 + y_3 + y_4 + y_5 + y_6 \leq 0$	$-x_2 + x_7 + y_3 + y_4 + y_5 + y_6 \leq 0$
$-x_2 + y_3 + y_4 + y_5 \leq 0$	$-x_2 + y_3 + y_4 + y_5 \leq 0$
$-x_1 + x_{12} - x_{14} + y_9 + y_{10} \leq 0$	$-x_1 + x_{12} - x_{14} + y_9 + y_{10} \leq 0$
$-x_1 + x_7 + x_{10} - x_{14} + y_6 + y_9 \leq 0$	$-x_{10} + x_{12} + y_{10} \leq 0$
$-x_1 + x_{10} - x_{14} + y_9 \leq 0$	$-x_2 - x_{10} + x_{12} + y_3 + y_4 + y_{10} \leq -1$
$-x_{12} + x_{14} + y_{11} + y_{12} \leq 0$	$-x_1 - x_{10} + x_{12} + y_2 + y_3 + y_4 + y_5 + y_{10} \leq -1$
$-x_{12} + y_{11} \leq 0$	$-x_2 - x_{10} + x_{12} + y_3 + y_4 + y_5 + y_{10} \leq -1$
	$-x_{12} + x_{14} + y_{11} + y_{12} \leq 0$
	$-x_{12} + y_{11} \leq 0$

References

Al-Ajeli, A., & Bordbar, B. (2016). Fourier-Motzkin method for failure diagnosis in Petri net models of discrete event systems. In *Proceedings of the 13th international workshop on discrete event systems* (pp. 165–170). Xi'an, China.

Al-Ajeli, A., & Parker, D. (2018). Online fault diagnosis in Petri net models of discrete-event systems using Fourier-Motzkin. In *2018 UKACC 12th international conference on control* (pp. 397–402). http://dx.doi.org/10.1109/CONTROL.2018.8516748.

Basile, F. (2014). Overview of fault diagnosis methods based on Petri net models. In *2014 European control conference* (pp. 2636–2642). IEEE.

Basile, F., Chiacchio, P., & De Tommasi, G. (2009). An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, *54*(4), 748–759.

Basile, F., Chiacchio, P., & De Tommasi, G. (2012). On K-diagnosability of Petri nets via integer linear programming. *Automatica*, *48*(9), 2047–2058.

Basile, F., Chiacchiot, P., & Tommasi, G. D. (2008). Sufficient conditions for diagnosability of Petri nets. In *2008 9th international workshop on discrete event systems* (pp. 370–375). http://dx.doi.org/10.1109/WODES.2008.4605974.

Cabasino, M. P., Giua, A., Marcias, L., & Seatzu, C. (2012). A comparison among tools for the diagnosability of discrete event systems. In *2012 IEEE international conference on automation science and engineering* (pp. 218–223). IEEE.

Cabasino, M. P., Giua, A., Pocci, M., & Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems. *Control Engineering Practice*, *19*(9), 989–1001.

Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, *46*(9), 1531–1539.

Chouchane, A., Declerck, P., Khedher, A., & Kamoun, A. (2020). Diagnostic based on estimation using linear programming for partially observable petri nets with indistinguishable events. *International Journal of Systems Science: Operations & Logistics*, *7*(2), 192–205.

Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). Linear inequalities and polyhedra. In *Integer programming* (pp. 85–128). Springer.

Dantzig, G. B. (1972). *Fourier-Motzkin elimination and its dual*: *Tech. rep.*, DTIC Document.

Dotoli, M., Fanti, M. P., Mangini, A. M., & Ukovich, W. (2009). On-line fault detection of discrete event systems by Petri nets and integer linear programming. *Automatica*, *45*(11), 2665–2672.

Duffin, R. (1974). On Fourier's analysis of linear inequality systems. In M. Balinski (Ed.), *Mathematical programming studies*: *Vol. 1*, *Pivoting and extension* (pp. 71–95). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/BFb0121242.

Fanti, M. P., Mangini, A. M., & Ukovich, W. (2013). Fault detection by labeled Petri nets in centralized and distributed approaches. *IEEE Transactions on Automation Science and Engineering*, *10*(2), 392–404.

Jiroveanu, G., Boel, R. K., & Bordbar, B. (2008). On-line monitoring of large Petri net models under partial observation. *Discrete Event Dynamic Systems*, *18*, 323–354.

Kohler, D. A. (1967). *Projections of convex polyhedral sets*: *Tech. rep.*, DTIC Document.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, *77*(4), 541–580. http://dx.doi.org/10.1109/5.24143.

Pugh, W. (1991). The Omega test: A fast and practical integer programming algorithm for dependence analysis. In *Proceedings of the 1991 ACM/IEEE conference on supercomputing* (pp. 4–13). ACM.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555–1575.

Tsuji, K., & Murata, T. (1993). On reachability conditions for unrestricted Petri nets. In *Circuits and systems, 1993. 1993 IEEE international symposium on* (pp. 2713–2716). IEEE.

Wang, Y., Yin, L., & Zhu, G. (2020). Online fault diagnosis of labeled Petri nets based on reachability graphs and topological sorting. *IEEE Access*, *8*, 162363–162372.

Williams, H. P. (1976). Fourier-Motzkin elimination extension to integer programming problems. *Journal of Combinatorial Theory. Series A*, *21*(1), 118–123.

Zaytoon, J., & Lafortune, S. (2013). Overview of fault diagnosis methods for discret event systems. *Annual Reviews in Control*, *37*, 308–320.

Zhu, G., Feng, L., Li, Z., & Wu, N. (2020). An efficient fault diagnosis approach based on integer linear programming for labeled Petri nets. *IEEE Transactions on Automatic Control*, 1. http://dx.doi.org/10.1109/TAC.2020.3008712.

Zhu, G., Li, Z., & Wu, N. (2018). Model-based fault identification of discrete event systems using partially observed Petri nets. *Automatica*, *96*, 201–212.

Ahmed Al-Ajeli received the B.Sc. and M.Sc. degrees in Computer Science from the University of Babylon, Iraq, in 1999 and 2002, respectively. He worked as an assistant lecturer at the Department of Computer Science, the University of Babylon. In 2017, he received his Ph.D. in Computer Science from the University of Birmingham, the UK. Currently, he holds an Assistant Professor position at College of Information Technology, University of Babylon. His current research interests include fault diagnosis/prognosis in discrete-event systems, machine learning and anomaly detection.

David Parker is a Professor of Computer Science at the University of Birmingham. Prior to that he worked as a researcher at the University of Oxford. His main research interests are in formal verification, with a particular focus on the analysis of probabilistic systems. He has published over 140 papers in this area and was co-winner of the 2016 HVC award. He also leads the development of the widely used probabilistic verification tools PRISM and PRISM-games