Fault Model Identification and Synthesis in Petri Nets

Maria Paola Cabasino, Alessandro Giua, Christoforos N. Hadjicostis, Carla Seatzu*

July 15, 2014

Abstract

Fault identification studies in the Discrete Event Systems literature are typically model-based and require knowledge of the structure of the system, including the nature (and behavior) of the possible faults. In this paper we consider this problem within the framework of Petri nets assuming knowledge of the nominal (fault-free) system model but removing the requirement that the nature (or behavior) of the faults is known. Specifically, we consider a setting where faults are unobservable and use sequences of observations to infer the structure and behavior of faults. The resulting method recognizes the structure of the faulty system using knowledge of the structure of the fault-free system, and the projection of the faulty system language on the set of non-faulty events, which are assumed to be observable. Two problem formulations can be given: (i) fault identification when the resulting faulty Petri net system is required to generate all observed sequences, while no constraint is imposed on sequences that are not observed; (ii) fault synthesis where the resulting faulty Petri net system is required to *only* generate all observed sequences, while all sequences that are not observed cannot actually occur. We show that a solution to the first problem can always be easily found, while the synthesis problem is not trivial at all and we solve it via an approach based on linear integer programming, which allows us to take into account physical constraints on the system in terms of possible and not possible interactions in the system.

Published as:
M.P. Cabasino, A. Giua, C.N. Hadjicostis, C. Seatzu, "Fault Model Identification and Synthesis

*This work falls under the Cyprus Research Promotion Foundation (CRPF) Framework Programme for Research, Technological Development and Innovation 2009-2010 (CRPF's FP 2009–2010), co-funded by the Republic of Cyprus and the European Regional Development Fund, and specifically under Grant $T\Pi E/OPIZO/0609(BE)/08$. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of CRPF. This work has been partially supported by RAS project (L.R. n. 7/2007, Year 2010). M.P. Cabasino gratefully acknowledges Sardinia Regional Government for the financial support of her Post Doc fellowship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.).

M.P. Cabasino, A. Giua and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy. (E-mail: {cabasino,giua,seatzu}@diee.unica.it). Alessandro Giua is also with Aix-Marseille University, LSIS, France.C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, and also with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. (E-mail: chadjic@ucy.ac.cy)

in Petri Nets", *Discrete Event Dynamic Systems*, DOI: 10.1007/s10626-014-0190-x. The original publication is available at www.link.springer.com.

1 Introduction

In the discrete-event system (DES) framework, the data collected from the observation of the system is usually given in terms of behavioral descriptions (e.g., transition system and language) and the set of behavioral sequences may be fixed or may be increased in the course of the system operation (e.g. due to new experiments). The *identification* problem aims to address two main issues. First, it determines whether for the given behavioral specification there exists a DES (e.g., a Petri net of a given class or size) that generates the specified behavior. Second, it provides a constructive procedure to obtain a suitable DES model (usually an automaton or a Petri net).

Synthesis is a problem related to identification. While only a partial description of the system is assumed for identification, the synthesis problem starts from a complete behavioral description of the system. This does not imply that bisimulation is necessarily a goal to achieve, but a requirement of exactness or approximation needs to be specified as an input parameter of the synthesis problem. In simple words, the essential differences between identification and synthesis can be summarized in the following two items.

- While in many synthesis approaches there are both examples and counterexamples, i.e., respectively sequences of events that can be generated by the system and sequences of events that cannot be generated by the system, identification approaches typically do not consider counterexamples.
- In identification approaches only a fraction of all possible behaviors is observed, i.e., the system may allow for many traces in addition to the ones that are observed.

It is important to highlight that the above distinction between synthesis and identification is not universally recognized in the literature. Thus, it often occurs that the two terms are used interchangeably [6].

In this paper we deal with the problems of identification and synthesis of the fault model of a Petri net whose nominal behavior is known[1]. The proposed approach is based on some of our earlier results [8] where, given the language of a Petri net system, we identify the Petri net structure and its initial marking by solving an integer programming problem (IPP). We assume knowledge of the fault-free system and our goal is to identify the structure of the faulty part of the system, specifically the additional transitions that comprise the faulty behavior. We address both the identification and the synthesis problem. We consider faults as unobservable transitions, which implies that identification should only be based on the projection of the faulty system language on the set of non-faulty (observable) events.

[1] A preliminary and partial version of this paper was presented in [7].

We prove that a trivial solution to the identification problem can always be found even if such a solution may not in general have a physical meaning. Such a trivial solution, that consists by adding a fault transition with no input place, comes from the fact that the identification problem simply requires additional sequences of observable transitions to fire, while no disabling constraint should be met. On the contrary, the synthesis problem requires the simultaneous satisfaction of enabling and disabling constraints. A solution to the synthesis problem is proposed that is based on integer linear programming. Such a solution takes advantage of a linear algebraic characterization of the set of admissible solutions to the identification problem.

Our paper is based on the results we obtained in [8]. In particular both papers deal with identification and synthesis problems and both approaches are based on integer programming. However, the problem addressed in this paper cannot be seen as a particular case of the one in [8]. In fact we only have to identify (or synthesize) a part of the net, but the behavior of such a part is not observable. As a result of this requirement, the set of enabling and disabling constraints is significantly different with respect to [8] as discussed in the rest of the paper.

To the best of our knowledge very few contributions exist in the literature that deal with the problem of fault synthesis. On the contrary a very rich literature exists on the problem of identification and synthesis of discrete event systems modeled via Petri nets. The first solutions to the synthesis/identification problem of PNs date back to the early nineties and are based on the *theory of regions*. Most contributions along this line of research have been proposed by people from the computer science area. Among these we mention the works by Ehrenfeucht and Rozenberg [14], Badouel *et al.* [1], Badouel and Darondeau [2], and Cortadella *et al.* [11]. A series of more recent contributions based on the theory of regions have been brought by Bergenthum *et al.* [5], Carmona *et al.* [9], Lorenz *et al.* [15], and Lorenz and Juhás [16]. Independently, a series of other contributions have been proposed by people from the automatic control community. Among these we mention the works by Meda-Campaña and López-Mellado [17, 18], Cabasino *et al.* [8], Dotoli *et al.* [12] and Basile *et al.* [3]. An exhaustive survey on identification and synthesis of DES has been recently published by some of the authors [6].

The paper with the largest points of contact with this manuscript is one quite recently published by Dotoli *et al.* [13], which addresses the problem of identification of a Petri net system by modeling the unobservable behavior of a discrete event system. Assuming that the structure of the PN that captures the observable system behavior is known, they characterize the PN that models the unobservable system behavior. The PN system is recursively obtained by an on-line algorithm that detects the unobservable event occurrences and defines and solves (in some cases) a corresponding integer linear programming problem. The main difference between our approach and the approach in [13] is that they assume knowledge at each step of the marking of the Petri net system, while in our case only the firing of the observable transitions is given. The extra information (knowledge of PN marking) that is available in the approach in [13] also allows it to identify unobservable events that do not alter the nominal system behavior.

2 Background on Petri nets

In this section we introduce the formalism used in the paper. For more details on Petri nets, we refer the reader to [19].

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where $P = \{p_1, \ldots, p_m\}$ is a set of m places; $T = \{t_1, \ldots, t_n\}$ is a set of n transitions; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre*– and *post*–incidence functions, where $\mathbb{N}$ is the set of nonnegative integers, that specify the arcs; we use $C = Post - Pre$ to denote the incidence matrix of N. Pictorially, places are represented by circles, and transitions by bars.

The *preset* of a place (resp., transition), denoted by ${}^\bullet p$ (resp., ${}^\bullet t$), is the set of input transitions (resp., places), i.e., ${}^\bullet p = \{t \in T \mid Post(p,t) > 0\}$ (${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$). The *postset* of a place (resp., transition), denoted by $p^\bullet$ (resp., $t^\bullet$), is the set of output transitions (resp., places), i.e., $p^\bullet = \{t \in T \mid Pre(p,t) > 0\}$ ($t^\bullet = \{p \in P \mid Post(p,t) > 0\}$).

A *marking* is a vector $M : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a nonnegative integer number of tokens, represented by black dots. We use $M(p)$ to denote the marking of place p. A *P/T system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition $t \in T$ is enabled at M if $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t)$. We write $M\ [\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at M, and we write $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'. We denote the length of the firing sequence σ by $|\sigma|$.

Given a sequence $\sigma \in T^*$, we call $\pi : T^* \rightarrow \mathbb{N}^n$ the function that associates with σ a vector $y \in \mathbb{N}^n$, called the *firing vector* of σ. We denote the ith entry of the firing vector $y = \pi(\sigma)$ by $y(i)$, and we denote the number of occurrences of transition t in sequence σ by $|\sigma|_t$, i.e., $y(i) = |\sigma|_{t_i}$.

A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0 \rangle$ and is denoted by $R(N, M_0)$.

Given a Petri net system $\langle N, M_0 \rangle$, we define its *language* as the set of firing sequences that are enabled at its initial marking M_0, $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$. We also define the set of firing sequences of length less than or equal to $k \in \mathbb{N}$ as $L_k(N, M_0) = \{\sigma \in L(N, M_0) \mid |\sigma| \leq k\}$. A language $\mathcal{L}$ is said to be *prefix-closed* if for any string $\sigma \in \mathcal{L}$, all prefixes of σ are in $\mathcal{L}$.

Given a nominal net system $\langle N, M_0 \rangle$, it is possible to associate with it a faulty net whose set of transitions $T^F = T \cup T_f$ contains the set of observable transitions T of the nominal net, plus a set of unobservable fault transitions T_f. We define the *projection operator* $P_o : (T^F)^* \rightarrow T^*$ recursively as follows: (i) $P_o(\varepsilon) = \varepsilon$, where ε is the empty word ; (ii) $P_o(t) = t \quad \forall t \in T$; (iii) $P_o(f) = \varepsilon \quad \forall f \in T_f$; (iv) $P_o(\sigma t) = P_o(\sigma) P_o(t) \quad \forall \sigma \in (T^F)^*, t \in T^F$. We denote as ${P_o}^{-1}$ the inverse of the projection operator P_o. Note that ${P_o}^{-1}$ returns a set.

We also define the *k-projection operator* $P_{o,k} : (T^F)^* \rightarrow \cup_{i=0}^{k} T^k$ as the restriction of the operator

P_o to only those sequences that lead to a projected word of length less than or equal to k.

The projection over the set of unobservable and fault transitions T_f is denoted P_u.

Finally, given a net $N = (P, T, Pre, Post)$, and a subset $T' \subseteq T$ of its transitions, we define the T'*−induced subnet of* N as the new net $N' = (P, T', Pre', Post')$ where $Pre', Post'$ are the restrictions of $Pre, Post$ to T'. The net N' can be obtained from N by removing all transitions in $T \setminus T'$.

3 Problem formulation

In the sequel two different problems are proposed. In both cases we assume that:

(A1) The behavior of the fault-free system is known and fault occurrences never forbid sequences that are enabled in the nominal behavior.

However, it may happen that some other sequences of events that were not allowed in the nominal behavior, become enabled after the faults occurrence.

In the first problem formulation, called *identification*, we simply want to be sure that all the observations performed after the fault occurrence are actually enabled by the faulty model. On the contrary, in the second problem formulation, called *synthesis*, given a finite set of observed words, we want to be sure that such words completely describe the observable behavior of the faulty system, in the sense that all observable words that do not belong to such a given set, are actually forbidden.

Let $\langle N, M_0 \rangle$ be the known net system that generates a nominal (i.e., *fault-free*) language $\mathcal{L}$; both the net structure $N = (P, T, Pre, Post)$ and the language $\mathcal{L}$ are known. The set of transitions T of the nominal net is composed by all *observable* transitions. We consider a *faulty* net system $\langle N^F, M_0 \rangle$, where $N^F = (P, T^F, Pre^F, Post^F)$, that has the same number of places and the same initial marking as the nominal one, but its set of transitions is $T^F = T \cup T_f$, where $T_f = \{f_1, \ldots, f_q\}$ is the set of fault transitions. Furthermore, we make the following assumption.

(A2) The pre– and post–incidence matrices of the faulty net are

$$Pre^F = \left[\begin{array}{cccc} Pre & Pre^{f_1} & \cdots & Pre^{f_q} \end{array} \right],$$

$$Post^F = \left[\begin{array}{cccc} Post & Post^{f_1} & \cdots & Post^{f_q} \end{array} \right],$$

where Pre^{f_i} (resp., $Post^{f_i}$) is the $m \times 1$ pre–incidence (resp., post–incidence) matrix of transition f_i.

According to assumption (A2), the faulty net retains the structure of the nominal one but includes a number of additional fault transitions. Note that the number of fault transitions is not known a priori. One can assume a certain number (based on the knowledge of the nominal system) and look for the solution of the identification/synthesis problem. If the integer programming problem has no solution, then the number of fault transitions should be increased. Numerical examples presented in Subsection 5.5 help to understand how the number of fault transitions can be chosen.

3.1 Case I: fault model identification

Problem 3.1 *Let us consider a fault-free net system $\langle N, M_0 \rangle$. Let $\mathcal{L}_k^F \subseteq T^*$ be a finite prefix-closed language over T whose strings have length less than or equal to a given integer k.*

We want to identify a faulty net system $\langle N^F, M_0 \rangle$ satisfying (A1) and (A2) and such that for each $\sigma \in \mathcal{L}_k^F$ it holds ${P_o}^{-1}(\sigma) \cap L(N^F, M_0) \neq \emptyset$. ■

In simple words, our goal is to identify the structure of the faulty system, based on the knowledge of its observed language, namely the projection of its firing sequences (that include fault transitions) over the set of observable transitions T. Note that, since we are solving an identification problem, the language $L(N^F, M_0)$ may also contain sequences that do not belong to the set ${P_o}^{-1}(\mathcal{L}_k^F)$ (and thus produce a sequence of observations that does not belong to $\mathcal{L}_k^F$).

Example 3.2 *Consider the fault-free net system in Fig. 1(a), whose language is $\mathcal{L} = \{(t_1t_2)^n \mid n \geq 0\} \cup \{(t_1t_2)^n t_1 \mid n \geq 0\}$. Assume that a fault f may occur, and that the observable language of the system with faults having length smaller than or equal to $k = 2$ is $\mathcal{L}_2^F = \{\varepsilon, t_1, t_2, t_1t_2\}$. We want to identify a net system that coincides with the net system in Fig. 1(a) if the fault transition and its connected arcs are removed, and whose language projected on $\{t_1, t_2\}$ and restricted to only sequences of length smaller than or equal to $k = 2$ includes all sequences in $\mathcal{L}_2^F$.*

One solution to this is given by the net system in Fig. 1(b); however, this is not the only possible solution. Thus, we have to associate an appropriate performance index to select a solution, within the set of admissible ones, that best matches some given criteria. Note that, for the Petri net in Fig. 1(b) we have $L(N^F, M_0) = \{\varepsilon, t_1, f, t_1t_2, t_1f, ft_1, ft_2, ff, t_1t_2t_1, t_1t_2f, t_1ft_2, t_1ff, ft_1t_2, ft_1f, ft_2t_1, ft_2f, fft_1, fft_2, fff, \ldots\}$ that strictly includes ${P_o}^{-1}(\mathcal{L}_2^F) = \{\varepsilon, t_1, f, t_1t_2, t_1f, ft_1, ft_2, ff, t_1t_2f, t_1ft_2, t_1ff, ft_1t_2, ft_1f, ft_2f, fft_1, fft_2, fff, \ldots\}$ but also includes other strings; e.g. $ft_2t_1 \notin {P_o}^{-1}(\mathcal{L}_2^F)$ but $ft_2t_1 \in L(N^F, M_0)$. ■

The next proposition characterizes the existence of a solution for this problem.

Proposition 3.3 Given a fault-free system $\langle N, M_0 \rangle$, a necessary and sufficient condition for the existence of a solution to Problem 3.1 is that $L_k(N, M_0) \subseteq \mathcal{L}_k^F$.

Proof: The necessity follows from the fact that assumptions (A1) and (A2) guarantee that all sequences firable in $\langle N, M_0 \rangle$ can also be fired in $\langle N^F, M_0 \rangle$. Thus Problem 3.1 is well-posed only if $\mathcal{L}_k^F$ contains all the sequences in $L_k(N, M_0)$.

The sufficiency of the condition follows from the fact that a trivial solution to Problem 3.1 may

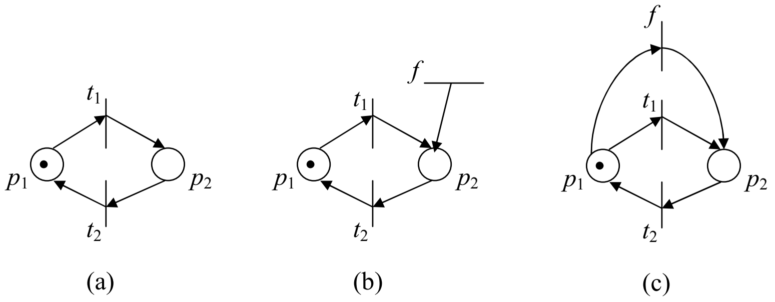

Figure 1: A motivational example.

also be computed by simply adding to the non-faulty net one fault transition with no input place and connected to all places that are input places for the transitions of T. Such a transition is always enabled and may fire as many times as necessary to enable all words in $L_k(N, M_0)$. □

The above proposition clearly makes evident that a trivial solution to the identification problem always exists that consists of a source fault transition, i.e., a transition with no input places, that may fire as many times as necessary to enable the sequences in $\mathcal{L}_k^F$ that are not in $L_k(N, M_0)$. Obviously, such a solution is in general not significant in real applications.

Note that, depending on the cost function that we use, we can select one solution or another. As an example if we choose a cost function that minimizes the number of fault transitions and the arcs weight we would obtain a solution with a unique fault transition having a set of post arcs in those places that need more tokens to enable a word in $\mathcal{L}_k^F \setminus \mathcal{L}_k$.

In the rest of the paper we assume that the condition $L_k(N, M_0) \subseteq \mathcal{L}_k^F$ stated in Proposition 3.3 holds.

The following section formalizes the synthesis problem. Note that for such a problem computing a solution is not trivial. Indeed, in this case we cannot simply enable additional sequences that were forbidden with no fault, but we also have to simultaneously disable other sequences.

3.2 Case II: fault model synthesis

Problem 3.4 *Consider a fault-free net system $\langle N, M_0\rangle$ and let $\mathcal{L}_k^F \subseteq T^*$ be a finite prefix-closed language over T whose strings have length less than or equal to a given integer k.*

We want to synthesize a faulty net system $\langle N^F, M_0\rangle$ satisfying (A1) and such that

$$P_{o,k}(L(N^F, M_0)) = \mathcal{L}_k^F.$$

■

In such a case, since we are solving a synthesis problem, the net needs to generate exactly the observable sequences of length less than or equal to k that are contained in $\mathcal{L}_k^F$.

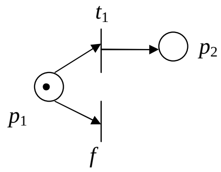

Figure 2: A Petri net system where $\mathcal{L} = \mathcal{L}_1^F = \{\varepsilon, t_1\}$.

Example 3.5 *Consider again the fault-free net system in Fig. 1(a), whose language is* $\mathcal{L} = \{(t_1t_2)^n \mid n \geq 0\} \cup \{(t_1t_2)^n t_1 \mid n \geq 0\}$. *Assume that a fault* f *may occur, and the observable language of the system with faults is* $\mathcal{L}^F = \{((t_1+\varepsilon)\ t_2)^n \mid n \geq 0\} \cup \{((t_1+\varepsilon)\ t_2)^n t_1 \mid n \geq 0\}$. *We want to synthesize a net system that retains the structure of the fault-free Petri net in Fig. 1(a), and whose language projected on* $T = \{t_1, t_2\}$ *is equal to* $\mathcal{L}^F$. *The Petri net in Fig. 1(b) is no longer an acceptable solution. A solution to this problem is given by the net system in Fig. 1(c), but again this is not the only possible solution. Thus, we have to associate an appropriate performance index to select a solution, within the further set of admissible solutions, that best matches some given criteria. This issue will be discussed further in the following sections.* ■

Note that the necessary condition provided by Proposition 3.3 for the identification problem also applies to the synthesis problem. However, in the case of synthesis such a condition is obviously no longer sufficient.

As a final remark, note that for both identification and synthesis we can only reconstruct faults generating strings whose observable projection is not contained in the language of the nominal system. The following example clarifies this.

Example 3.6 *Let us consider the net system in Fig. 2, where* $T = \{t_1\}$ *and* $T_f = \{f\}$. *Here* $k = 1$ *and* $\mathcal{L} = \mathcal{L}_1^F = \{\varepsilon, t_1\}$, *i.e., the nominal language coincides with the observable language of the faulty system. This means that after the firing of fault transition* f *no anomalous string will be observed, thus this fault cannot be identified. In fact after the fault occurrence only the the empty word* ε *will be observed. This, however, can be explained by a nominal evolution where transition* t_1 *has not (yet) fired.* ■

A possible solution for problems such as the one in the previous example may be offered by the notion of *forcible* events. The idea is that an operator can force the occurrence of an *observable* event, i.e., it can make it occur (assuming it is enabled). In the previous example, if no string is observed the operator may force event t_1: if it fires, one can infer that f has not occurred, while if it does not fire, one may infer that it has been disabled by the firing of f.

4 A linear algebraic characterization of the solutions of the fault identification problem

In this section we solve Problem 3.1 using the approach in [8]. We make the following assumption that holds for the rest of the paper.

(A3) The T_f-induced subnet is *acyclic.*

This assumption is justified by the fact that our approach is based on the incidence matrix and on the state equation of the net. As a consequence of Assumption (A3), all fault transitions are loop-free so the incidence matrix contains all the information on the net structure. Moreover, it guarantees necessary and sufficient conditions for reachability in the unobservable subnet.

In this section we provide an algebraic characterization of the set of admissible faulty systems. In particular, we show that if an upper bound is given on the number of times each fault transition may fire, then the characterization is linear.

4.1 Preliminary characterization

Definition 4.1 *Let $\mathcal{L}$ be the prefix-closed language of the fault-free net system $\langle N, M_0 \rangle$, and $\mathcal{L}_k^F$ be the projection over the set of observable transitions T of the prefix-closed language of the faulty net system we want to identify.*

We define the following sets:

$$\begin{array}{rcl} \mathcal{E} & = & \{(\sigma, t_j) \mid \sigma \in \mathcal{L},\ |\sigma| < k,\ t_j \in T,\ \sigma t_j \in \mathcal{L}\}, \\ \mathcal{E}^F & = & \{(\sigma, t_j) \mid \sigma \in \mathcal{L}_{k-1}^F,\ t_j \in T,\ \sigma t_j \in \mathcal{L}_k^F\}, \\ \tilde{\mathcal{E}}^F & = & \mathcal{E}^F \setminus \mathcal{E}. \end{array} \tag{1}$$

■

In simple words, $\mathcal{E}$ includes all couples (σ, t_j) where σ is in the language $\mathcal{L}$ of the fault-free net system, with length smaller than k, and transition $t_j \in T$ such that the sequence σt_j also belongs to the language $\mathcal{L}$ of the fault-free net system.

The set $\mathcal{E}^F$ includes all couples (σ, t_j) where σ is in the projection of the language of the faulty net system on the set of observable transitions T, with length smaller than k, and transition $t_j \in T$ such that the sequence σt_j also belongs to the projection of the language of the faulty net system on the set of observable transitions T.

Clearly, it is $\mathcal{E} \subseteq \mathcal{E}^F$. Therefore the set $\mathcal{E}^F \setminus \mathcal{E}$ contains only those couples (σ, t_j) in $\mathcal{E}^F$ that originate from the occurrence of some fault.

Proposition 4.2 *Consider a pair* $(\sigma, t_j) \in \tilde{\mathcal{E}}^F$. *Under assumption (A3), the faulty net* $\langle N^F, M_0\rangle$ *generates a word* $(\sigma t_j)^F \in {P_o}^{-1}(\sigma t_j)$ *such that* $|(\sigma t_j)^F|_{f_i} = \alpha^i_{\sigma,j}$, $i = \{1, \dots, q\}$, *iff the following conditions are both verified:*

(a) The net $\langle N^F, M_0\rangle$ *generates a word* $\sigma^F \in {P_o}^{-1}(\sigma)$ *with* $|\sigma^F|_{f_i} = \alpha^i_\sigma$, *where* $i = \{1, \dots, q\}$.

(b) For each fault transition f_i, *with* $i = \{1, \dots, q\}$, *there exists integer* $\alpha^i_{\sigma,j}$ *such that*

$$\begin{cases} M_0 + \sum_{i=1}^q \alpha^i_{\sigma,j}(Post^{f_i} - Pre^{f_i}) + C \cdot \pi(\sigma) \geq Pre(\cdot, t_j) \\ \alpha^i_{\sigma,j} \geq \alpha^i_\sigma, \quad \forall i \in \{1, \dots, q\}. \end{cases} \tag{2}$$

Proof. (*If* part) If the net $\langle N^F, M_0\rangle$ generates a word whose projection is σt_j, then there exists a firing sequence

$$M_0[\sigma^F\rangle M[\nu\rangle M'[t_j\rangle,$$

where $\sigma^F \in {P_o}^{-1}(\sigma)$ and $\nu \in T_f^*$, where $|\nu|_{f_i} = \alpha^i_j$, with $\alpha^i_j \geq 0$ and $i \in \{1, \dots, q\}$, are additional occurrences of f_i, that may be necessary to enable transition t_j after σ^F has fired. Let $|\sigma^F|_{f_i} = \alpha^i_\sigma$; then according to the state equation it holds

$$\begin{aligned} M' &= M_0 + C \cdot \pi(\sigma) + \textstyle\sum_{i=1}^q \alpha^i_\sigma \cdot (Post^{f_i} - Pre^{f_i}) \\ &\quad + \textstyle\sum_{i=1}^q \alpha^i_j \cdot (Post^{f_i} - Pre^{f_i}) \\ &= M_0 + C \cdot \pi(\sigma) + \textstyle\sum_{i=1}^q \alpha^i_{\sigma,j} \cdot (Post^{f_i} - Pre^{f_i}) \end{aligned}$$

with $\alpha^i_{\sigma,j} = \alpha^i_\sigma + \alpha^i_j$, with $i \in \{1, \dots, q\}$, and, since M' enables t_j, we obtain (2).

(*Only if* part) Assume condition (a) is verified so that there exists a marking M such that $M_0[\sigma^F\rangle M$. This allows us to rewrite (2) as

$$\begin{cases} M + \sum_{i=1}^q \alpha^i_j \cdot (Post^{f_i} - Pre^{f_i}) \geq Pre(\cdot, t_j) \\ \alpha^i_j \geq 0, \quad \forall i = \{1, \dots, q\} \end{cases}$$

where $\alpha^i_j = \alpha^i_{\sigma,j} - \alpha^i_\sigma$ for any $i \in \{1, \dots, q\}$. Consider now the unobservable subnet obtained from N by removing all transitions except f_i, with $i = \{1, \dots, q\}$, with initial marking M. By assumption (A3) the unobservable subnet is acyclic, hence inequality

$$M + \sum_{i=1}^q \alpha^i_j \cdot (Post^{f_i} - Pre^{f_i}) \geq Pre(\cdot, t_j) \geq \vec{0}$$

implies that there exists a sequence of fault transitions, and a marking M' such that $M[\nu\rangle M'[t_j\rangle$ [10]. This means that a sequence $(\sigma t_j)^F \in {P_o}^{-1}(\sigma t_j)$ is firable in the faulty net with $|(\sigma t_j)^F|_{f_i} = \alpha^i_{\sigma,j} = \alpha^i_\sigma + \alpha^i_j$, for all $i \in \{1, \dots, q\}$. □

4.2 IPP formulation

In this section we provide a linear algebraic characterization of the set of net systems that satisfy a given identification problem. Note that, as already discussed earlier, a solution to the identification problem can be easily found by adding fault source transitions that pump an appropriate number of tokens to enable all sequences in $\mathcal{L}^F \setminus \mathcal{L}$. However, the importance of a linear algebraic characterization is twofold: first it is useful if an appropriate cost function is given, second it provides the starting point for the synthesis problem addressed later on.

Proposition 4.3 *Consider the following set of algebraic constraints:*

$$
\mathcal{G}^{id}(\tilde{\mathcal{E}}^F) \triangleq
\left\{
\begin{array}{l}
\left.\begin{array}{l}
M_0 + \sum_{i=1}^{q} \alpha^i_{\sigma,j} \cdot (Post^{f_i} - Pre^{f_i}) + C \cdot \pi(\sigma) \geq Pre(\cdot, t_j) \\
\alpha^i_{\sigma,j} \in \mathbb{N} \\
Post^{f_i}, Pre^{f_i} \in \mathbb{N}^m \\
\forall\ (\sigma, t_j) \in \tilde{\mathcal{E}}^F,\ \forall i \in \{1, \ldots, q\}
\end{array}\right\} (a) \\
\\
\left.\begin{array}{l}
Pre^{f_i}(p_k) - z^i_k \cdot K \leq 0 \\
Post^{f_i}(p_k) - (1 - z^i_k) \cdot K \leq 0 \\
z^i_k \in \{0, 1\} \\
\forall i \in \{1, \ldots, q\}, \quad \forall k \in \{1, \ldots, m\}
\end{array}\right\} (b)
\end{array}
\right.
\qquad (3)
$$

whose unknowns are $\alpha^i_{\sigma,j}$, $\forall\ (\sigma, t_j) \in \tilde{\mathcal{E}}^F$ and $\forall i \in \{1, \ldots, q\}$; $Post^{f_i}, Pre^{f_i} \in \mathbb{N}^m$, $\forall i \in \{1, \ldots, q\}$; the binary variables z^i_k $\forall i \in \{1, \ldots, q\}$ and $\forall k \in \{1, \ldots, m\}$. Finally, K is a very large constant, greater than the largest admissible value of the weight of any pre and post arc associated with fault transitions.

Any couple of matrices Pre^F and $Post^F$ satisfying such a set of constraints constitutes a solution of Problem 3.1, provided that the resulting T_f induced subnet is acyclic, namely assumption (A3) is satisfied.

Proof: *We first prove that, under assumptions (A1) to (A3), the net system $\langle N^F, M_0 \rangle$ satisfies $P_k(L(N^F, M_0)) \supseteq \mathcal{L}^F_k$ if and only if the set of algebraic constraints in (3) holds.*

Provided that the T_f induced subnet is acyclic, therefore state equation describes all traces, constraints (a) are the enabling constraints *relative to those sequences that can only be observed when faults occur, i.e., a transition t_j is enabled at $M_0 + \sum_{i=1}^{q} \alpha^i_{\sigma,j} \cdot (Post^{f_i} - Pre^{f_i}) + (Post - Pre) \cdot \pi(\sigma)$ if and only if $M_0 + \sum_{i=1}^{q} \alpha^i_{\sigma,j} \cdot (Post^{f_i} - Pre^{f_i}) + (Post - Pre) \cdot \pi(\sigma) \geq Pre \cdot \pi(t_j)$. They trivially follow from Proposition 4.2. Indeed, by Proposition 4.2, for any couple $(\sigma, t_j) \in \tilde{\mathcal{E}}^F$, there exists a couple $(\sigma', t'_j) \in \tilde{\mathcal{E}}^F$ such that $\sigma' t'_j = \sigma$ (e.g. if $(t_1, t_2) \in \tilde{\mathcal{E}}^F$ then also $(\varepsilon, t_1) \in \tilde{\mathcal{E}}^F$).*

Constraints (b) force each fault transition f_i to be loop-free. In fact, they imply that if $Pre^{f_i}(p_k) > 0$, then $Post^{f_i}(p_k) = 0$, and viceversa. □

As mentioned in the statement of Proposition 4.3 the resulting T_f induced subnet has to be acyclic. Thus once we obtain a solution, we need to check for the acyclicity of the T_f induced subnet, and if this condition is not respected we need to add some constraint to avoid such a solution (e.g imposing one pre or post arc of the T_f induced subnet obtained as a solution equal to zero) and run again the simulation.

4.3 Constraints linearization

The nonlinearity of constraints (3), due to the product of $\alpha^i_{\sigma,j}$ and $Post^{f_i}, Pre^{f_i}$, can be removed by assigning an upper bound Γ_i, for each fault transition f_i, on the number of times the fault transition f_i must fire to justify the anomalous behavior[2]. Obviously, Γ_i can be chosen differently for each fault transition f_i.

Let $\vec{c}_i$, $i = 1, \ldots, q$, be q m-dimensional vectors of integer numbers, each one associated with a different fault. Constraint (a) for the generic couple $(\sigma, t_j) \in \tilde{\mathcal{E}}^F$ can be translated into an OR constraint that can be written as the following set of $1+\sum_{i=1}^q [2(\Gamma_i+1)+1]+1 = 2+2\sum_{i=1}^q \Gamma_i+2q$ linear constraints involving mq integer variables and $\sum_{i=1}^q \Gamma_i + q$ binary variables:

$$\left\{\begin{array}{l} M_0 + \sum_{i=1}^q \vec{c}_i + C \cdot \pi(\sigma) - Pre(\cdot, t_j) \geq 0 \\ \left.\begin{array}{l} \vec{c}_i - h(Post^{f_i} - Pre^{f_i}) \leq z_{i,h}\vec{K} \\ \vec{c}_i - h(Post^{f_i} - Pre^{f_i}) \geq -z_{i,h}\vec{K} \\ \sum_{h=0}^{\Gamma_i} z_{i,h} = \Gamma_i \\ z_{i,h} \in \{0,1\}, \quad \vec{c}_i \in \mathbb{Z}^m \end{array}\right\} \qquad i = 1, \ldots, q, \quad h = 0, 1, \ldots, \Gamma_i \\ \sum_{i=1}^q z_{i,0} \geq 1 \end{array}\right. \tag{4}$$

where, as usual, K is a very large constant (see [4] for a description of this procedure to convert OR constraints into a conjunction of linear ones), and $\vec{K} = K \cdot \vec{1}_m$. In simple words, we define an m-dimensional vector $\vec{c}_i = h(Post^{f_i} - Pre^{f_i})$ for each fault $i \in \{1, \ldots, q\}$. The first constraint means that provided that a sufficiently large number of faults occur interleaved with σ, then transition t_j is enabled in the faulty net system. The second and third constraint in (4) are redundant if $z_{i,h} = 1$ while they impose that $\vec{c}_i = h(Post^{f_i} - Pre^{f_i})$ if $z_{i,h} = 0$. The forth constraint means that we have Γ_i constraints of the previous form for each fault class i. The fifth constraint specifies that $z_{i,h}$ are binary variables, while $\vec{c}_i$ are integer vectors. Finally the sixth constraint imposes that at least one fault has to occur to enable t_j after σ.

We conclude this section by noting that it may happen that the set of constraints (4) is infeasible. This obviously means that the values of the Γ_i's have not been selected sufficiently large, since, as already discussed earlier, the identification problem always has a solution.

4.4 Analysis of number of constraints and unknowns in the linear set (4)

Let n be the cardinality of T, k the length of the longest string in $\mathcal{L}^F_k$, and ν_r, for $r = 0, \ldots, k$, the number of strings in $\mathcal{L}^F_k \setminus \mathcal{L}$ of length r.

Then the nonlinear constraint set (3) contains

[2] Note that a tradeoff should be made when choosing Γ_i: a large value of Γ_i makes the linearization less restrictive but results in higher computational complexity. We assume here that a tentative value of Γ_i is initially taken, and it is then increased if the resulting set of linear constraints is infeasible.

- $m \sum_{r=1}^{k} \nu_r$ constraints of type (a),
- $2 \cdot m \cdot q$ constraints of type (b).

When linearized, the number of constraints (a) becomes equal to

$$m \cdot \left(2 + 2 \sum_{i=1}^{q} \Gamma_i + 2q \right) \cdot \sum_{r=1}^{k} \nu_r .$$

The total number of unknowns in the nonlinear set is equal to

$$u_{\text{nl-id}} \;=\; 2mq + q \cdot \sum_{r=1}^{k} \nu_r + mq = 3mq + q \cdot \sum_{r=1}^{k} \nu_r ,$$

where the right-side terms in the first equality are due, respectively, to the number of pre– and post–incidence arcs for all fault transitions f_i with $i = 1, \ldots, q$, the integer variables $\alpha^i_{\sigma,j}$ in (a), and the binary variables z^i_k in (b), with $i = 1, \ldots, q$ and $k = 1, \ldots, m$.

The total number of unknowns in the linearized set is

$$u_{\text{l-id}} \;=\; 2mq + \left((m+1)q + \sum_{i=1}^{q} \Gamma_i \right) \cdot \sum_{r=1}^{k} \nu_r + mq.$$

The total number of unknowns in the nonlinear set in the worst case is

$$u_{\text{nl-id}_{\text{MAX}}} \leq 3mq + q \cdot \textstyle\sum_{r=1}^{k} n^r = \mathcal{O}(mq + qn^k),$$

and, considering $\bar{\Gamma} = \max(\Gamma_i)$, for all $i = 1, \ldots, q$, the total number of unknowns in the linear set in the worst case is

$$u_{\text{l-id}_{\text{MAX}}} \leq 3mq + ((m+1)q + q\bar{\Gamma}) \cdot \textstyle\sum_{r=1}^{k} n^r = \mathcal{O}(((m+1)q + q\bar{\Gamma}) \cdot n^k),$$

i.e., this problem has exponential complexity with respect to k.

5 Fault synthesis

In this section we focus on Problem 3.4. The proposed solution is based on a linear algebraic characterization of the set of possible solutions of the synthesis problem that can be seen as an extension of the results in Section 4. Indeed in this case we assume to have complete knowledge of the language of the faulty system at least for observable words of length less than or equal to a given integer k, thus we need to consider not only examples but also counterexamples. Assumptions (A1)–(A3) still hold.

5.1 Preliminary characterization

Definition 5.1 *Let $\mathcal{L}$ be the prefix-closed language of the fault-free net system, and $\mathcal{L}_k^F$ be the prefix-closed observed language of the faulty net system we want to identify.*

In addition to the sets in equation (1)*, we define the set*

$$\tilde{\mathcal{D}}^F \quad = \quad \{(\sigma, t_j) \mid \sigma \in \mathcal{L}_{k-1}^F,\ t_j \in T,\ \sigma t_j \notin \mathcal{L}_k^F\}. \tag{5}$$

■

In simple words, $\tilde{\mathcal{D}}^F$ includes all couples (σ, t_j) where σ is the projection of a string in the language of the faulty net system on the set of observable transitions T, with length at most equal to $k-1$, and t_j is a transition in T such that σt_j does not belong to the projection of the language of the faulty net system on the set of observable transitions T.

Definition 5.2 *Let (σ, t_j) be a pair in $\tilde{\mathcal{D}}^F$. We define Υ_σ as the set of the minimum firing vectors of fault transitions necessary to enable σ:*

$$\Upsilon_\sigma = \{\vec{\gamma}_\sigma \in \mathbb{N}^q \ \mid \ \pi(P_u(\sigma^F)) = \vec{\gamma}_\sigma \wedge \nexists\, \vec{\gamma}'_\sigma \lneq \vec{\gamma}_\sigma, \text{where}\ \pi(P_u(\sigma^F)) = \vec{\gamma}'_\sigma\ \text{and}\ \sigma^F \in {P_o}^{-1}(\sigma)\}. \tag{6}$$

■

The characterization given in Proposition 4.2 still holds. Let us now give a characterization for the set $\tilde{\mathcal{D}}^F$.

Proposition 5.3 *Under assumption (A3) the faulty net $\langle N^F, M_0\rangle$ disables a transition t_j after all sequences $\sigma^F \in {P_o}^{-1}(\sigma)$ that are enabled at M_0, iff $\exists \vec{\gamma}_\sigma \in \Upsilon_\sigma$ such that $\forall\ \vec{\gamma}_M \in \mathbb{N}^q$, with $\vec{\gamma}_M \geq \vec{\gamma}_\sigma$, it holds*

$$M_0 + C \cdot \pi(\sigma) + \sum_{i=1}^{q} \gamma_M(i) \cdot (Post^{f_i} - Pre^{f_i}) \ngeq Pre(\cdot, t_j). \tag{7}$$

Proof. Let us show the *if* part. As well known, a transition t is not enabled at a marking $M' \in R(N, M_0)$ iff $M' \ngeq Pre(\cdot, t)$.

Now, if t_j is not enabled after the firing of all sequences $\sigma^F \in {P_o}^{-1}(\sigma)$ at M_0, then $\forall\ \vec{\gamma}^F = \pi(P_u(\sigma^F))$ it should be

$$M_0 + C \cdot \pi(\sigma) + \textstyle\sum_{i=1}^{q} \gamma^F(i) \cdot (Post^{f_i} - Pre^{f_i}) \ngeq Pre(\cdot, t_j),$$

or, equivalently, equation (7) should be verified for all $\vec{\gamma}_M \geq \vec{\gamma}_\sigma$, where the set of $\vec{\gamma}_\sigma$ is defined as in equation (6).

Let us now prove the *only if* part. Since the unobservable subnet is acyclic, the state equation gives conditions that are necessary and sufficient for the reachability (and for non-reachability as well) [19]. Thus, if equation (7) is satisfied for all $\vec{\gamma}_M \in \mathbb{N}^q$, with $\vec{\gamma}_M \geq \vec{\gamma}_\sigma$, then it means that any marking M such that $M_0[\sigma^F\rangle M$ satisfies $M \ngeq Pre(\cdot, t_j)$. □

5.2 IPP formulation

Proposition 5.4 *Let us consider Problem 3.4 under assumptions (A1)–(A3), and let*

$$g(Pre^F, Post^F) = \sum_{s=1}^{m}\sum_{i=1}^{q}\Big[b_{s,i}Pre^{f_i}(p_s,\cdot) + c_{s,i}Post^{f_i}(p_s,\cdot)\Big]$$

be a given linear performance index, where $b_{s,i},\ c_{s,i} \in \mathbb{R}_0^+$, $s = 1, \ldots, m$, $i = 1, \ldots, q$.

A solution that minimizes $g(Pre^F, Post^F)$ *can be computed by solving the following nonlinear (IPP)*

$$\begin{cases} \min & g(Pre^F, Post^F) \\ s.t. & \mathcal{G}^{syn}(\tilde{\mathcal{E}}^F, \tilde{\mathcal{D}}^F) \text{ holds for the given } M_0 \end{cases} \tag{8}$$

where

$$\mathcal{G}^{syn}(\tilde{\mathcal{E}}^F, \tilde{\mathcal{D}}^F) \triangleq \left\{ \begin{array}{l} \mathcal{G}^{id}(\tilde{\mathcal{E}}^F) \\ \\ \left.\begin{array}{l} -KS^{f_i}_{\sigma,j} + M_0 + C\cdot\pi(\sigma) \\ \quad + \sum_{i=1}^{q}\gamma_M(i)\cdot(Post^{f_i} - Pre^{f_i}) - Pre(\cdot,t_j) \leq -\vec{1}_m \\ \vec{1}^{\,T} S^{f_i}_{\sigma,j} \leq m-1, \quad \forall i \in \{1,\ldots,q\} \\ S^{f_i}_{\sigma,j} \in \{0,1\}^m, \quad \forall i \in \{1,\ldots,q\} \\ \forall(\sigma,t_j) \in \tilde{\mathcal{D}}^F \\ \vec{\gamma}_M \in \mathbb{N}^q \end{array}\right\} (c) \\ \end{array} \right. \tag{9}$$

where $\mathcal{G}^{id}(\tilde{\mathcal{E}}^F)$ *is defined as in (3) and K (as usual) is a very large constant [4].*

Proof: *We already proved in Proposition 4.3 that, under assumptions (A1)–(A3), the net system* $\langle N^F, M_0\rangle$ *satisfies* $P_{o,k}(L(N^F, M_0)) \supseteq \mathcal{L}^F_k$ *if and only if the set of algebraic constraints in (3) holds.*

Thus, we only need to prove that, under assumptions (A1) and (A2), the net system $\langle N^F, M_0\rangle$ *satisfies* $P_{o,k}(L(N^F, M_0)) \subseteq \mathcal{L}^F_k$ *if and only if the set of algebraic constraints in (9) holds. To do this, we need to show that constraints (c) disable all those sequences of length less than or equal to k that do not belong to* $\mathcal{L}^F_k$.

Constraints (c) are the disabling constraints *relative to those sequences that are not enabled even if a fault transition occurs. They follow from Proposition 5.3 and their equivalence to constraints*

$$\begin{cases} M_0 + C\cdot\pi(\sigma) + \sum_{i=1}^{q}\gamma_M(i)\cdot(Post^{f_i} - Pre^{f_i}) \ngeq Pre(\cdot,t_j) \\ \forall(\sigma,t_j) \in \tilde{\mathcal{D}}^F \\ \forall\vec{\gamma}_M \in \mathbb{N}^q \end{cases}$$

can be proved as follows.

We first observe that, if t_j is not enabled at $M_0 + C \cdot \pi(\sigma) + \sum_{i=1}^{q} \gamma_M(i) \cdot (Post^{f_i} - Pre^{f_i})$, then there exists at least one place $p \in P$ such that

$$M_0(p) + C(p,\cdot) \cdot \pi(\sigma) + \textstyle\sum_{i=1}^{q} \gamma_M(i) \cdot (Post^{f_i}(p,\cdot) - Pre^{f_i}(p,\cdot)) \leq Pre(p,t_j) - 1. \qquad (10)$$

This holds for all p such that $S_{\sigma,j}^{f_i}(p) = 0$, where $S_{\sigma,j}^{f_i}$ is a binary vector having as many entries as the number of places of the net. Having $\vec{1}^T S_{\sigma,j}^{f_i} \leq m - 1$ this implies that this occurs for at least one place $p \in P$. Note that the set of constraints (c) disable the sequence σt_j even when all fault transitions f_i with $i \in \{1,\ldots,q\}$ fire $\gamma_M(i)$ times. For this reason the binary vector $S_{\sigma,t_j}^{f_i}$ has the upper index f_i.

Finally, we observe that assuming $\vec{\gamma}_M \in \mathbb{N}^q$ in (c) rather than $\vec{\gamma}_M \geq \vec{\gamma}_\sigma$ (see equation (7)), introduces no spurious markings. In fact, by definition of $\vec{\gamma}_\sigma$, constraints (c) are redundant for all $\gamma_M(i) \in [0, \gamma_\sigma(i))$, with $i \in \{1,\ldots,q\}$. □

Two remarks need to be made concerning the above IPP formulation.

- It is reasonable to assume that in several real applications it is known a priori that a fault that may affect a given subnet, has no effect on some parts of the net. In such a case it is sufficient to impose that some entries in the Pre^F and $Post^F$ matrices are null, thus also reducing the number of unknowns.
- The second remark concerns the performance index $g(Pre^F, Post^F)$ that assigns different weights to arcs. In particular, a high weight associated with a given arc is equivalent to assuming a low probability of having such an arc in the faulty system. On the contrary, small weights are associated with those arcs that are more likely to appear in the faulty system, e.g., on the basis of some a priori information on the considered system and/or some considerations on its layout. As a particular case, we may look for a solution that assumes a small number of additional fault transitions. In such a case, we can assign a unitary weight to all entries in the first column of matrices Pre^F and $Post^F$. Then, we can assign a weight equal to a given $\alpha > 1$ to all entries in the second column of such matrices. The weight assigned to arcs in the third column can be taken equal to α^2 and so on.

5.3 Constraints linearization

The set of constraints (9) that are necessary to characterize the set of admissible solutions are nonlinear (see constraints (a) and (c)). In Subsection 4.3 we already presented a way to linearize constraints (a). In this subsection, we present a way to remove the nonlinearity of the disabling constraints.

In this case the nonlinearity can also be removed by assigning to each fault transition f_i an upper bound Γ_i on the number of times the fault transition f_i may fire. This upper bound Γ_i is the same as the one used to linearize constraints (a) in Subsection 4.3. Constraint (c) for the generic couple $(\sigma, t_j) \in \tilde{\mathcal{D}}^F$ can be translated into an AND constraint that can be written as follows in terms of linear constraints:

$$\left\{\begin{array}{l} -KS^{f_i}_{\sigma,j} + M_0 + C \cdot \pi(\sigma) + \sum_{i=1}^{q} \lambda_i \cdot (Post^{f_i} - Pre^{f_i}) - Pre(\cdot, t_j) \leq -\vec{1}_m \\ \vec{1}^{\,T} S^{f_i}_{\sigma,j} \leq m-1 \end{array}\right. \qquad (l2)$$

where λ_i is a natural number that can take values from 0 to Γ_i, i.e., $\lambda_i = 0, 1, \ldots, \Gamma_i$. Note that we have to write $\Pi_{i=1}^{q}(\Gamma_i + 1) - 1$ constraints of the form $(l2)$ where we consider all possible combinations of values of λ_i, except for the one where all λ_i's are equal to zero.

An important remark needs to be made concerning the linearization of the disabling constraints. Indeed it may happen that a solution to the linerarized set of constraints is found for a given set of Γ_i's. However, if there exists some fault f_i that may actually fire a number of times larger than Γ_i, it may occur that the resulting net system violates some of the disabling constraints, i.e., the net obtained solving the linearized set of constraints violates the nonlinear constraints. To be sure that disabling constraints are satisfied we need to limit the number of times a fault transition can fire. Thus, once the linearization is performed and the upper bound on the number of firings each fault transition may fire to satisfy the enabling and disabling constraints is known, an input place to each fault transition should be added having as initial marking the upper bound associated to the fault transition.

5.4 Complexity of the synthesis procedure

Since the set of constraints (9) is a superset of the set of constraints (3) and for such a set we have already computed the number of constraints and unknowns in Subsection 4.4, we just need to evaluate the number of constraints and unknowns coming from constraints (c) in (9).

Let n be the cardinality of T, k the length of the longest string in $\mathcal{L}_k^F$, and ν_r (ν'_r, resp.), for $r = 0, \ldots, k$, the number of strings in $\mathcal{L}_k^F \setminus \mathcal{L}$ ($\mathcal{L}_k^F$, resp.) of length r.

Then the number of nonlinear constraints (c) in (9) is equal to

$$(m+1) \sum_{r=0}^{k-1} (n \cdot \nu'_r - \nu'_{r+1}).$$

The way to see this is as follows. The faulty language $\mathcal{L}_k^F$ contains a certain number ν_r of strings of length r. The number of strings of length $r+1$ that have to be disabled is equal to the number of strings of length r in $\mathcal{L}_k^F$ multiplied by the number of transitions n *minus* the number of strings of length $r+1$ in $\mathcal{L}_k^F$. Since for each string we have $m+1$ disabling constraints (c) (where m is the number of places of the Petri net), the total number of constraints is as given above.

When linearized, the number of constraints (c) becomes equal to

$$(m \cdot \Pi_{i=1}^{q}[(\Gamma_i + 1) - 1] + 1) \sum_{r=0}^{k-1} (n \cdot \nu'_r - \nu'_{r+1}).$$

The total number of unknowns in the nonlinear IPP (9) is

$$\begin{aligned} u_{nl-syn} &= u_{nl-id} + mq\sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1}) \\ &= 3mq + q \cdot \sum_{r=1}^{k} \nu_r + mq\sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1}), \end{aligned}$$

where the last term of the right-side terms is due to the binary vectors $S^{f_i}_{\sigma,j}$, with $i = 1, \dots, q$.

The total number of unknowns in the linear IPP is

$$\begin{aligned} u_{l-syn} &= u_{l-id} + mq{\cdot}\Pi_{i=1}^{q}\Gamma_i{\cdot}\sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1}) \\ &= 3mq + ((m+1)q + \sum_{i=1}^{q}\Gamma_i) \cdot \sum_{r=1}^{k} \nu_r + mq \cdot \Pi_{i=1}^{q}\Gamma_i \cdot \sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1}). \end{aligned}$$

Note that for given values of k and n, it is possible to find a worst case bound for $\rho = \sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1})$. In fact, it holds:

$$\begin{aligned} \rho &= \sum_{r=0}^{k-1}(n \cdot \nu'_r - \nu'_{r+1}) \\ &= n \cdot \nu'_0 + (n-1) \cdot \left(\sum_{r=1}^{k-1} \nu'_r\right) - \nu'_k \\ &= n + (n-1) \cdot \left(\sum_{r=1}^{k-1} \nu'_r\right) - \nu'_k. \end{aligned}$$

This expression is maximized if we assume $\nu'_k = 0$ while all other ν'_r take the largest possible value, i.e., $\nu'_r = n^r$. Hence, we have

$$\rho \leq n + (n-1) \cdot (n + \cdots + n^{k-1}) = n^k,$$

so that the total number of unknowns in the nonlinear IPP in the worst case is

$$u_{\text{nl-syn}_{\text{MAX}}} \leq 3mq + q \cdot \textstyle\sum_{r=1}^{k} n^r + qm \cdot n^k = \mathcal{O}(m\ q\ n^k),$$

and, considering $\bar{\Gamma} = max(\Gamma_i)$, for all $i = 1, \dots, q$, the total number of unknowns in the linear IPP in the worst case is

$$u_{\text{l-syn}_{\text{MAX}}} \leq 3mq + ((m+1)q + q(\bar{\Gamma})) \cdot \textstyle\sum_{r=1}^{k} n^r + mq\bar{\Gamma}^q \cdot n^k = \mathcal{O}(m\ q\ \bar{\Gamma}^q\ n^k),$$

i.e., this problem has exponential complexity with respect to q and k.

5.5 Numerical examples

In this section we present two examples. First, we provide an example to better explain the proposed procedure, then we discuss the problem of acyclicity and the necessity of Assumption (A3).

Example 5.5 *Let us consider the net in Fig. 3(a) that models a simple manufacturing system. The complexity of the example is limited on purpose to clearly illustrate the procedure and to show how physical information on the system layout can be easily taken into account to obtain a realistic faulty model.*

Place p_1 models a buffer containing parts that have not been correctly assembled. Such parts need to be first disassembled and this corresponds to the firing of transition t_1. As a result of the disassembly process, 4 parts of type A, 1 part of type B and 1 part of type C are obtained. Such parts are put in a new buffer where a machine process them before being assembled again. In particular, such operations on parts of type A (B and C, respectively) are modeled by place p_2 (p_3 and p_4, respectively), while the assembly operation corresponds to the firing of transition t_2. At this point, parts are put in a buffer (place p_5) and a final operation, e.g., cleaning or painting, is performed, modeled by transition t_3.

Assuming as in Fig. 3(a) that only one part is initially present in the first buffer, the language representing the regular behavior of the system is $\mathcal{L} = \{\varepsilon, t_1, t_1t_2, t_1t_2t_3\}$.

Now, let $\mathcal{L}_4^F = \{\varepsilon, t_1, t_1t_2, t_1t_2t_3, t_1t_2t_2, t_1t_2t_2t_2\}$, i.e., sequences $t_1t_2t_2$ and $t_1t_2t_2t_2$ also become firable. We have $\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_2), (t_1t_2, t_3)\}$, $\mathcal{E}^F = \{(\varepsilon, t_1), (t_1, t_2), (t_1t_2, t_3), (t_1t_2, t_2), (t_1t_2t_2, t_2)\}$, $\tilde{\mathcal{E}}^F = \{(t_1t_2, t_2), (t_1t_2t_2, t_2)\}$ and $\tilde{\mathcal{D}}^F = \{(\varepsilon, t_2), (\varepsilon, t_3), (t_1, t_1), (t_1, t_3), (t_1t_2, t_1), (t_1t_2t_2, t_1), (t_1t_2t_2, t_3)\}$.

If we look for a solution that only assumes the presence of one fault transition and that minimizes the arc weights associated with the fault transition, we obviously get the solution where f is a transition with no input places, one output arc to p_3 and one output arc to p_4. However, such a solution has clearly no physical meaning, since we cannot assume infinite capacity buffers connected to places p_3 and p_4.

If we impose that at least one input place to f exists, e.g., given

$$Pre^F = [Pre^F(p_1, f) \ \ldots \ Pre^F(p_5, f)]^T$$

we impose that $\sum_{i=1}^{5} Pre^F(p_i, f) \geq 1$, we get the solution in Fig. 3(b).

Note however, that such a solution may also not be realistic because it implies that from one part coming from p_2 we get two parts, one in p_3 and one in p_4. To avoid this we can impose the additional constraint: $\sum_{i=1}^{5} Pre^F(p_i, f) = \sum_{i=1}^{5} Post^F(p_i, f)$. In such a case we get the solution in Fig. 3(c) where the weight of the input arc to f is set equal to 2.

If the same problem is solved assuming two fault transitions the net in Fig. 3(b) is still a solution. Another possible solution is the faulty net system in Fig. 3(d) that is probably the most realistic solution if we assume that operations on parts of type A, B and C are performed on adjacent

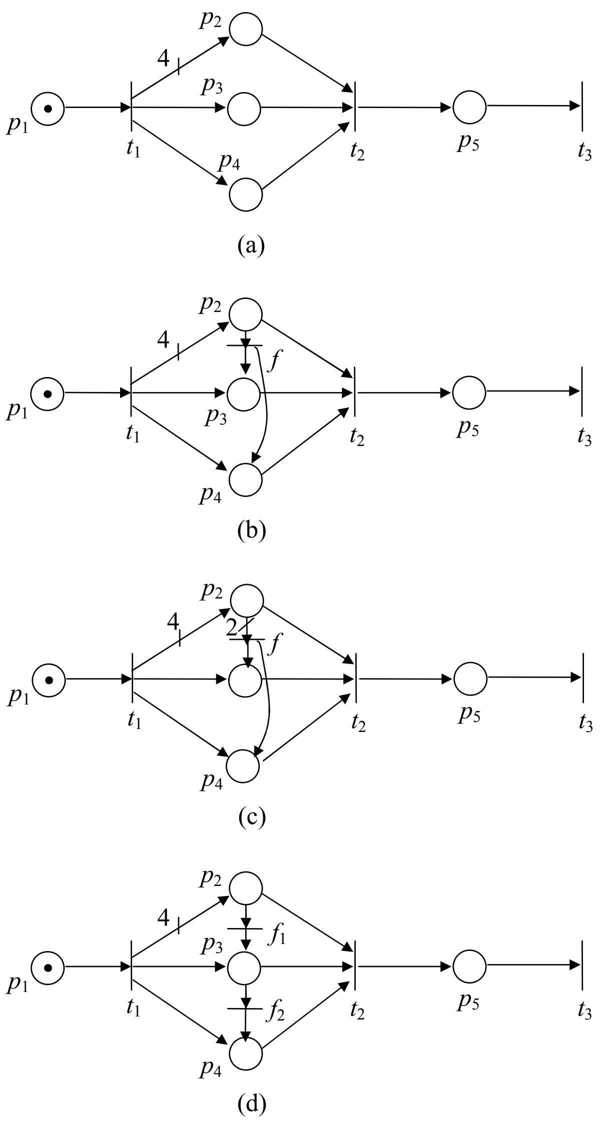

Figure 3: (a) The fault-free net system, and (b), (c) and (d) three different faulty net systems.

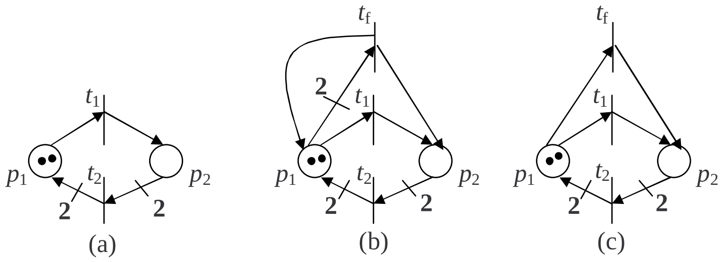

Figure 4: (a) The fault-free net system of Example 5.6, (b) the faulty net system where t_f is not loop-free, (c) the equivalent net of the one represented in (b).

lines, so it is unlikely that one part (and even more two) are dropped from the first to the third line. Also such a constraint can be easily imposed using: $Pre^F(p_2, f_2) = 0$. Note that the solution shown in Fig. 3(d) can be obtained only imposing $\Gamma_1 \geq 2$ and $\Gamma_2 \geq 1$; in fact, with the net system in Fig. 3(d), transition f_1 has to fire twice to satisfy the set of constraints (9) *with $\tilde{\mathcal{E}}^F$ and $\tilde{\mathcal{D}}^F$ defined above.* ■

Example 5.6 *Let us consider the net in Fig. 4(a) and the two languages $\mathcal{L}_3 = \{\varepsilon, t_1, t_1t_1, t_1t_1t_2\}$ and $\mathcal{L}_3^F = \{\varepsilon, t_1, t_1t_1, t_1t_2, t_1t_1t_2, t_1t_2t_1\}$. Assume that we want to synthesize the Petri net system that minimizes the arc weights associated with the fault transition. Assume that the problem is solved assuming only one fault transition t_f. This requires the solution of a linearized IPP of the form* (9) *where $\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_1), (t_1t_1, t_2)\}$, $\mathcal{E}^F = \{(\varepsilon, t_1), (t_1, t_1), (t_1, t_2), (t_1t_1, t_2), (t_1t_2,\, t_1)\}$, $\tilde{\mathcal{E}}^F = \{(t_1, t_2), (t_1t_2, t_1)\}$, and $\tilde{\mathcal{D}}^F = \{(\varepsilon, t_2), (t_1t_1, t_1), (t_1t_2, t_2)\}$.*

We note that a solution for these two languages exists and is represented by the faulty net system in Fig. 4(b), but if we apply the identification procedure proposed we obtain no integer solution even if the constraints relative to the acyclicity of the fault transition, i.e., the constraints (b) in (3)*, are removed. Since our constraints are based on the incidence matrix, the two nets shown in Fig. 4(b) and Fig. 4(c) are equivalent as far as our procedure is concerned. The problem is that for the net in Fig. 4(c) the disabling constraint on the couple (ε, t_2) is not verified, since transition t_2 can be enabled at M_0 after t_f has fired twice.* ■

6 Conclusions and future work

We presented the problem of identification and synthesis of the faulty model of a Petri net whose nominal behavior is known. We started from our previous results where, given the language of a Petri net system, we identify the Petri net structure and its initial marking by solving an integer programming problem. We assume that the fault-free system is known and we want to identify the structure of the faulty part of the system, specifically the additional transitions that comprise the faulty behavior are unobservable. We addressed both the identification and the synthesis problem.

There is a number of future directions of our research in this topic. First, we would like to study how the computational complexity can be reduced when considering particular net structures. Second, we plan to apply this procedure to some real application examples, possibly depending on some parameters. Then, we plan to investigate the possibility of having a bound on the number of times a fault can occur in the cost criterion. Moreover, we plan to generalize the approach considering the case of some silent transitions in the nominal model. Finally, we plan to consider the case where the faulty net includes additional places and transitions.

References

[1] E. Badouel, L. Bernardinello, and P. Darondeau. Polynomial algorithms for the synthesis of bounded nets. *Lecture Notes in Computer Science*, 915:647–679, 1995.

[2] E. Badouel and P. Darondeau. Theory of regions. *Lecture Notes in Computer Science*, 1491:529–586, 1998.

[3] F. Basile, P. Chiacchio, J. Coppola, and G. De Tommasi. Identification of Petri nets using timing information. In *3rd Int. Workshop on Dependable Control of Discrete Systems*, Saarbrücken, Germany, 2011.

[4] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–429, 1999.

[5] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser. Synthesis of Petri nets from infinite partial languages. In *Proc. 8th Int. Conf. on Application of Concurrency to System Design*, XiŠan, China, 2008.

[6] M.P. Cabasino, P. Darondeau, M.P. Fanti, and C. Seatzu. Model identification and synthesis of discrete-event systems. *Contemporary Issues in System Science and Engineering*, IEEE/Wiley Press Book Series, 2014.

[7] M.P. Cabasino, A. Giua, C.N. Hadjicostis, and C. Seatzu. Fault model identification with Petri nets. In *Proc. 9th IFAC Work. on Discrete Event Systems*, Gotheborg, Sweden, 2008.

[8] M.P. Cabasino, A. Giua, and C. Seatzu. Identification of Petri nets from knowledge of their language. *Discrete Events Dynamic Systems*, 17(4):447–474, 2007.

[9] J. Carmona, J. Cortadella, A. Kishinevsky, L. Lavagno, A. Kondratyev, and A. Yakovlev. A symbolic algorithm for the synthesis of bounded Petri nets. In *Proc. Int. Conf. on Application and Theory of Petri Nets and Other Models of Concurrency*, Xian, China, 2008.

[10] D. Corona, A. Giua, and C. Seatzu. Marking estimation of Petri nets with silent transitions. *IEEE Trans. Automatic Control*, 52(9):1695–1699, 2007.

[11] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Deriving Petri nets from finite transition systems. *IEEE Transactions on Computers*, 47(8):859–882, 1998.

[12] M. Dotoli, M.P. Fanti, and A.M. Mangini. Real time identification of discrete event systems using Petri nets. *Automatica*, 44(5):1209–1219, 2008.

[13] M. Dotoli, M.P. Fanti, A.M. Mangini, and W. Ukovich. Identification of DES unobservable behaviour by Petri nets. In *Proc. 2nd IFAC Workshop on Dependable Control of Discrete Systems*, Bari, Italy, 2009.

[14] A. Ehrenfeucht and G. Rozenberg. Partial (Set) 2-Structures - Part 1 and Part 2. *Acta Informatica*, 27(4):315–368, 1989.

[15] R. Lorenz, G. Juhás, and S. Mauser. How to synthesize nets from languages – a survey. In *Proc. 2007 Winter Simulation Conference*, Washington DC, USA, 2007.

[16] R. Lorenz and G. Juhás. Towards synthesis of Petri nets from scenarios. *Lecture Notes in Computer Science*, 4024:302–321, 2006.

[17] M.E. Meda-Campaña and E. López-Mellado. Incremental synthesis of Petri net models for identification of discrete event systems. In *Proc. 41th IEEE Conf. on Decision and Control*, Las Vegas, Nevada USA, 2002.

[18] M.E. Meda-Campaña and E. López-Mellado. Required event sequences for identification of discrete event systems. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 3778–3783, Maui, Hawaii, USA, 2003.

[19] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.