Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-7 (2018) 374–380

Design of Monitor-based Supervisors in Labelled Petri Nets

Ziyue Ma [*] Zhou He [**] Zhiwu Li [***] Alessandro Giua [****]

[*] School of Electro-Mechanical Engineering, Xidian University, Xi'an, China (maziyue@gmail.com)
[**] College of Mechanical & Electrical Engineering, Shaanxi University of Science and Technology, Xi'an, China (hzakyhr@gmail.com)
[***] Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, and SEME, Xidian University, Xi'an, China (systemscontrol@gmail.com, zhwli@xidian.edu.cn)
[****] DIEE, University of Cagliari, Cagliari, Italy (giua@diee.unica.it).

Abstract: In this paper we propose a method to compute a monitor-based supervisor that enforces a GMEC in labelled Petri nets. Due to the presence of indistinguishable transitions, an observable GMEC may not be enforceable by a monitor place in general. To solve this problem we introduce the notion *dependency* of transitions that can be used to judge whether an event is generated by an input/output transition or its spurious ones, and we present an algorithm to design a monitor-based supervisor based on the notion of *Hilbert basis*. This approach is purely structural with low online computational cost. Moreover, our supervisor is robust to some change of initial markings or irrelevant part of the plant net.

Keywords: Supervisory Control, Petri Net, Discrete Event System

1. INTRODUCTION

Supervisory Control Theory (SCT), originated by Ramadge and Wonham Ramadge and Wonham (1989), provides a unifying framework for modeling and control of *discrete event systems* (DESs) and has been widely used in modeling various physical systems. Petri nets have been proposed as SCT models since they provide an efficient solution to control problems (e.g., *deadlock prevention* (Li and Zhou, 2004; Li et al., 2012; Nazeem and Reveliotis, 2015), *fault diagnosis* Cabasino et al. (2015), and *marking avoidance* (Moody and Antsaklis, 2000; Ma et al., 2015; Basile et al., 2015)). In Petri nets, a state specification consists in a set of legal markings, and the control objective consists in preventing the system from reaching the *forbidden markings*. There exists a rich literature on supervisory control of partially observable systems both using automata and Petri nets (Moody and Antsaklis, 2000; Luo et al., 2009; Basile et al., 2013; Ma et al., 2017a; Luo and Zhou, 2017). In many practical problems a state specification is given as a *generalized mutual exclusion constraint* (GMEC) (Giua et al., 1992) and can be efficiently implemented as a monitor place.

Although the aforementioned methods can handle cases with unobservable transitions, they implicitly require that any pair of observable transitions be distinguishable, i.e., the firing of each observable transition can be uniquely recognized. As we will see in Section III, however, these methods are not applicable if there are indistinguishable transitions.

In many real supervisory control systems the available sensors may not be able to uniquely detect all possible events. These systems are thus modeled by labelled Petri nets with indistinguishable transitions. In fact, many practical systems with a limit of sensors are modeled by *labelled Petri nets* with indistinguishable transitions. However, a GMEC can no longer be implemented as a monitor place in these nets.

To solve GMEC control in labelled Petri nets with indistinguishable transitions, supervisors based on *state estimation* are proposed. A supervisor of such type makes control decisions based on the computation of the set of consistent markings from the knowledge of the events generated by the plant. To reduce the computational load, state space abstraction techniques (Ru et al., 2014; Ma et al., 2017b) have been proposed. However, the computation cost to design a state-estimation-based supervisor is usually very high due to the offline construction of the observer. For example, the size of a basis reachability graph can be as large as the reachability graph.[1] Moreover, this type of supervisors is also highly dependent on the initial marking and the net structure, which means that the supervisor must be re-designed once the initial marking or the net structure/observation structure changes.

This paper aims to present a *monitor-based* supervisor for Petri nets that have indistinguishable transitions. This problem has not been previously addressed in literature, as far as we know. A monitor-based supervisor is a controller that takes control decisions by observing the occurrence events without performing additional complex computation such as state estimation. To simplify the presentation, in this paper we consider nets that are fully controllable, and the initial GMEC representing the control demand is *observable*, since given an uncontrollable/unobservable GMEC we can always[2] compute some controllable and observable GMECs (e.g., by the methods in

[⋆] This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61472295, 61703321, the Huashan Scholarship of ShaanXi Province, and the Science Technology Development Fund, MSAR, under Grant No. 078/2015/A3.

[1] Although it is possible to put the estimation procedure online in stepwise manner, in each iteration an online marking enumeration is still needed, which is also computationally heavy.

[2] To guarantee the control problem has a solution, it is generally assumed that the initial marking is admissible, i.e., at the initial marking the control demand is not violated by firing only uncontrollable transitions.

(Moody and Antsaklis, 2000; Ma et al., 2017a; Luo and Zhou, 2017)) satisfying the original control demand. We first show that in labelled nets an observable GMEC is in general not implementable by a monitor place. Moreover, a supervisor without global information may perform a very conservative control decision. To solve this problem, we introduce *dependent transitions* that can be used to identify that if an event is generated by an input/output transition of a GMEC or its spurious transitions with the same label. Finally we present an algorithm to design an online supervisor. Our method is based on the computation of the *Hilbert basis* and is based on a structural approach that does not require to enumerate the set of reachable markings. The computation of the Hilbert basis, although known to be NP-hard (Chubarov and Voronkov, 2005), is done during the offline stage, and hence the online computational load is low since the marking estimation is avoided. Moreover, it is robust with respect to the change of the initial marking and of the irrelevant part of the plant net.

The paper is organized in seven sections. Section 2 recalls the basic notions of labelled Petri nets and the supervisory control. Section 3 formalizes the problem. Section 4 proposes dependency of transitions and private subnets. In Section 5 a quantitative relation on transition dependency is given followed by an algorithm to design an online supervisor. Section 6 and Section 7 presents an example and draws the conclusions, respectively.

2. PRELIMINARIES

2.1 Petri Net

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P is a set of m *places* represented by circles; T is a set of n *transitions* represented by bars; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence functions* that specify the arcs in the net and are represented as matrices in $\mathbb{N}^{m\times n}$ (here $\mathbb{N} = \{0, 1, 2, \ldots\}$). The *incidence matrix* of a net is defined by $C = Post - Pre \in \mathbb{Z}^{m\times n}$ (here $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$).

For a transition $t \in T$ we define its *set of input places* as ${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$ and its *set of output places* as $t^\bullet = \{p \in P \mid Post(p,t) > 0\}$. The notion for ${}^\bullet p$ and $p^\bullet$ are analogously defined.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place of a Petri net a non-negative integer number of tokens, represented by black dots and can also be represented as a m component vector. We denote by $M(p)$ the marking of place p. A *marked net* $\langle N, M_0\rangle$ is a net N with an initial marking M_0. We denote by $R(N, M_0)$ the set of all markings reachable from the initial one.

A transition t is *enabled* at M if $M \geq Pre(\cdot,t)$ and may fire reaching a new marking $M' = M + C(\cdot,t)$. We write $M[\sigma\rangle M'$ to denote that the sequence of transitions $\sigma \in T^*$ is enabled at M and yields M'. We denote the set of all sequences firable from M_0 as $L(N, M_0)$, i.e., $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$. We use $\sigma(t) = k$ to denote that t occurs k times in σ. The vector $\mathbf{y}_\sigma$ is the Parikh vector of $\sigma \in T^*$, i.e., $y_\sigma(t) = \sigma(t) = k$ if transition t occurs k times in σ.

A sequence σ is called *repetitive* if $C \cdot \mathbf{y}_\sigma \geq \mathbf{0}$, since if $M[\sigma\rangle M'$ and $M' \geq M$ then $M[\sigma^r\rangle$ holds for any $r \in \mathbb{N}$. In other words, if σ can fire at a marking M, then it can fire infinite number of times from M.

A path from $x_0 \in P\cup T$ to $x_k \in P\cup T$ is a sequence of nodes $x_0x_1\cdots x_{k-1}x_k$ such that for all $i \in \{1,\ldots,k\}$, $x_{i-1} \in {}^\bullet x_i$ holds. A path from x_0 to x_k is said to be *simple* if $x_i \neq x_0$ and $x_i \neq x_k$ for all $i \in \{1,\ldots,k-1\}$.

2.2 Labelled Petri Nets

A *labelled Petri net* (LPN) is a 4-tuple $G = (N, M_0, E, \ell)$, where $\langle N, M_0\rangle$ is a marked net, E is the *alphabet* (a set of labels), and $\ell : T \to E \cup \{\varepsilon\}$ is the *labelling function* that assigns to each transition $t \in T$ either a symbol from E or the empty word ε. We use T_e to denote the set of transitions whose labels are all e, and T_t to denote the set $\{t\}$.

The set of transitions T can be partitioned into two disjoint sets $T = T_o \cup T_{uo}$, where $T_o = \{t \in T \mid \ell(t) \in E\}$ is the set of *observable* transitions and $T_{uo} = T \setminus T_o = \{t \in T \mid \ell(t) = \varepsilon\}$ is the set of *unobservable* transitions.

An LPN $G = (N, M_0, E, \ell)$ is said to be:

- *free-labelled* if $(t_1 \neq t_2) \to (\ell(t_1) \neq \ell(t_2))$ and for all $t \in T$, $\ell(t) \neq \varepsilon$;
- *λ-free labelled* if for all $t \in T$, $\ell(t) \neq \varepsilon$;
- *arbitrarily labelled* otherwise.

2.3 Supervisor

In a partially controllable LPN, the set of events E is partitioned into the *set of controllable events* E_c and the *set of uncontrollable events* E_{uc}, i.e., $E = E_c \cup E_{uc}$. This naturally leads to a partition on the set of transitions $T = T_c \cup T_{uc}$ where $T_c = \{t \in T \mid \ell(t) \in E_c\}$ is the *set of controllable transitions* and $T_{uc} = \{t \in T \mid \ell(t) \in E_{uc} \cup \{\varepsilon\}\}$ is the *set of uncontrollable transitions*. In this paper we assume that the net is fully controllable, i.e., $E_c = E$.

The objective of a control agent, i.e., the *supervisor*, is to ensure that only legal markings $\mathscr{L} \subset \mathbb{N}^m$ are reached by preventing firings that yield some illegal markings. A supervisor runs in parallel with the plant net and at each step makes a control decision from the knowledge of the observation so far. Given an observation $w \in E_o^*$ (* is the *Kleene star*), a control decision is made by allowing a set of events $Ctrl(w) \subseteq E_c$. Note that if the supervisor disables event e, all transitions labelled e are all disabled. In other words, a transition t is firable if it is currently enabled and $\ell(t) \in Ctrl(w)$.

2.4 GMECs

A *Generalized mutual exclusion constraint* (Giua et al., 1992) (GMEC) is a pair $(\mathbf{w}, k)$ where $\mathbf{w} \in \mathbb{Z}^m$ and $k \in \mathbb{Z}$. A GMEC defines a set of *legal markings*:

$$\mathscr{L}_{(\mathbf{w},k)} = \{M \in \mathbb{N}^m \mid \mathbf{w}^T \cdot M \leq k\}.$$

Given a GMEC $(\mathbf{w}, k)$ and a marking M, the quantity $\mathbf{w}^T \cdot M$ is called its *token count* at marking M.

Definition 1. Given a net N and a GMEC $(\mathbf{w}, k)$, a transition is said to be an *input (resp. output) transition* of $(\mathbf{w}, k)$ if $\mathbf{w}^T \cdot C(\cdot,t) > 0$ (resp. $\mathbf{w}^T \cdot C(\cdot,t) < 0$). The set of input (resp. output) transitions of $(\mathbf{w}, k)$ is denoted as $I_\mathbf{w}$ (resp. $O_\mathbf{w}$). □

Definition 2. (Monitor place). Giua et al. (1992); Moody and Antsaklis (2000) Given a net $\langle N, M_0\rangle$ and a GMEC $(\mathbf{w}, k)$, the monitor place of $(\mathbf{w}, k)$ is a place p_w with its incidence matrix and the initial marking:

$$\begin{cases} \forall t \in T, C(p_w,t) = -\mathbf{w}^T \cdot C(\cdot,t) \\ M_0(p_w) = k - \mathbf{w}^T \cdot M_0 \end{cases} \quad (1)$$

□

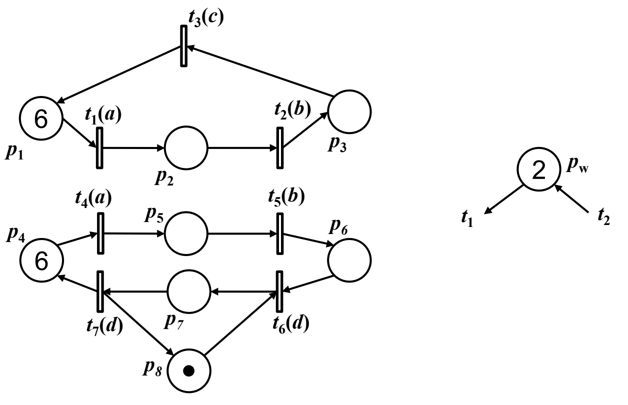

Fig. 1. (Left) A labelled Petri net, and (right) a monitor place for underlying unlabelled net.

3. OBSERVABLE GMECS

The control specifications in many practical cases can be described by GMECs. If the net is fully controllable and the net is free-labelled, i.e., each transition is distinguishable, a GMEC $(\mathbf{w},k)$ can be easily implemented by adding a loop-free monitor place p_w to the plant net as in Definition 2.

Example 1. Consider the Petri net in Figure 1 as a free-labelled net, i.e., all labels are ignored. Suppose that we want to enforce a GMEC $(\mathbf{w},k) = ([0,1,0,0,0,0,0,0]^T,2)$ such that the tokens in p_2 should not exceed 2. The supervisor can be represented as a monitor place p_w added to the plant net as shown on the right of the figure. □

In Example 1, the monitor runs like a counter that is initialized as $Sum = M_0(p_w) = 2$ and monitors the firings of t_1 and t_2 online: (1) each time it observes the firing of t_1 (resp. t_2), Sum decreases (resp. increases) by 1, and (2) while $Sum = 0$ it prevents any firing of t_1 until $Sum > 0$ by detecting of the firing of t_2. Such control policy can be easily obtained and implemented: it is sufficient to only monitor t_1 and t_2, i.e., it does not need any global information, and is robust with respect to the initial marking change and the modification of net structures that are irrelevant to t_1, t_2, and p_2.

A monitor may not implementable when some of the transitions are uncontrollable or when the net is partially observable. The issue of uncontrollability has been discussed in many papers (Moody and Antsaklis, 2000; Luo et al., 2009; Basile et al., 2013; Ma et al., 2017a; Luo and Zhou, 2017) and will not be considered here, i.e., we assume all transitions are controllable. Similarly, if the plant net is not free-labelled, the above method to enforce a GMEC becomes unfeasible. Notice that the monitor place p_w has arcs to and from each transition t with $\mathbf{w}^T \cdot C(\cdot,t) \neq 0$. This implies that at each step the supervisor must *exactly know* if such t has fired, in order to correctly update the tokens in p_w. However, in an arbitrarily labelled LPN this may not be possible due to two types of the nondeterminism:

(1) A transition t is *unobservable* if $\ell(t) = \varepsilon$, and hence the supervisor is not able to detect the firing of t;
(2) Two transitions t' and t'' are *indistinguishable* if $\ell(t') = \ell(t') = a$, and thus their firing produces the same observation. In this case by observing a the supervisor may not be able to determine whether t_1 or t_2 has fired;

To characterize the first type of nondeterminism we recall the *observability of GMECs*.

Definition 3. Moody and Antsaklis (2000) A GMEC $(\mathbf{w},k)$ is said to be *observable* if for all $t \in T_{uo}$, $\mathbf{w}^T \cdot C(\cdot,t) = 0$ holds. □

The physical meaning of observability of a GMEC is that its token count can change only due to the firing of observable transitions. The observability of GMEC guarantees that the monitor place p_w does not interact with unobservable transitions, and in this case the first type of nondeterminism is bypassed. To characterize the second type of nondeterminism, the following definition is necessary.

Definition 4. Given an LPN $G = (N,M_0,E,\ell)$, an observable GMEC $(\mathbf{w},k)$ is said to be *distinguishable* if:

$$(\mathbf{w}^T \cdot C(\cdot,t) \neq 0) \quad \Rightarrow \quad (\forall t' \neq t, \ell(t') \neq \ell(t)).$$

□

By Definition 4 a distinguishable GMEC is observable and each transition that modifies its token count is assigned a unique label. A distinguishable GMEC can be implemented by a monitor place p_w as that in Eq. (1), since the supervisor can uniquely detect the firing each transition which modifies the token count of the GMEC.

Given an arbitrary GMEC, methods in (Moody and Antsaklis, 2000; Luo et al., 2009; Ma et al., 2017a; Luo and Zhou, 2017) guarantee that the resulting GMEC(s) is observable. If all observable transitions are distinguishable, then the observability of a GMEC also implies its distinguishability. However, in nets with indistinguishable transitions an observable GMEC is not always distinguishable. On the other hand, we will shortly see that a GMEC that is not distinguishable cannot be implemented as a monitor place. To better characterize the second nondeterminism that hinders the monitor-based solution, we introduce the notion of *spurious transitions*.

Definition 5. Given an input/output transition t of $(\mathbf{w},k)$, a transition $t' \neq t$ is said to be a *spurious transition* of t if $\ell(t') = \ell(t)$ and $\mathbf{w}^T \cdot C(\cdot,t') = 0$. The set of spurious transitions of t is denoted as $S(t)$. □

Due to the limit of space, we introduce the following assumption.

Assumption 1. The GMEC to be implemented satisfies:

$$(\mathbf{w}^T \cdot C(\cdot,t) \neq 0) \Rightarrow (\forall t' \neq t, \ell(t') = \ell(t) : \mathbf{w}^T \cdot C(\cdot,t') = 0).$$

□

Assumption 1 means that two input/output transitions of $(\mathbf{w},k)$ cannot be undistinguishable, i.e., for each event e there exists at most one transition t labelled by e such that $\mathbf{w}^T \cdot C(\cdot,t) \neq 0$. This assumption is purely technical, since the method of this paper can also be generalized to cases in which a GMEC may have several input/output transitions sharing the same label.

Due to the presence of spurious transitions, although a monitor-based supervisor can detect an event which may be generated by the firing of an input/output transition t, it cannot precisely know if such firing is due to t or some spurious one, since it does not have the knowledge of the current marking. A correct (and possibly maximally permissive) control decision can be made if the supervisor is a state-estimation-based one, i.e., it has the knowledge of all possible current markings by performing the online *marking estimation*. However, as we have mentioned in the Introduction, the estimation is usually computationally heavy (even with some space abstraction techniques in Petri nets such as the basis reachability graph (Cabasino et al., 2010; Ma et al., 2017b)). Moreover, a state-estimation-based supervisor must be re-designed in case that the net structure or the initial marking is modified, even if the modification is minor.

On the other hand, consider a monitor-based supervisor that does not have any knowledge of the current marking. To ensure that the control aim is not violated it has to act as follows:

- whenever the supervisor observes an event e and there exists an input transition t labelled e, it assumes that t has fired;
- whenever the supervisor observes an event e labeling an output transition t with a non-empty set of spurious transitions $S(t)$, it assumes that a transition in $S(t)$ has fired;

However, as shown in the following example, such control policy is very conservative and may even cause deadlocks.

Example 2. Consider the plant in Figure 1 with labels, i.e., $\ell(t_1) = \ell(t_4) = a$, $\ell(t_2) = \ell(t_5) = b$, $\ell(t_3) = c$, and $\ell(t_6) = \ell(t_7) = d$. Suppose that we still want to enforce the GMEC $(\mathbf{w}, k)$ such that the tokens in p_2 should not exceed 2.

Since the supervisor does not have knowledge of the current marking (i.e., it does not perform a marking estimation), once event a occurs, the supervisor cannot determine whether t_1 or t_4 fires, and hence it has to assume that such a comes from t_1; on the other hand, once event b occurs, the supervisor cannot determine whether t_2 or t_5 fires, and hence it assumes that such b comes from t_5. This reasoning leads to a control policy such that the supervisor disables event a forever after it observes a twice, which makes the plant inevitably dead. □

In the next sections of this paper we propose a method to design a monitor-based supervisor, which enhance the permissiveness of such control policy. The method is purely based on structural analysis and hence requires a relatively low computational load and the resulting supervisor is robust with respect to the initial marking change.

4. DEPENDENCY OF TRANSITIONS

Suppose that in the plant LPN in Figure 1 we observe event a for $r \in \mathbb{N}$ times. If we notice that event d occurs for $q \in \mathbb{N}$ times, we can conclude that t_4 the spurious transition of t_1 has fired at least $\lceil q/2 \rceil$ times. Hence we know that t_1 has fired at most $r - \lceil q/2 \rceil$ times. On the other hand, we can conclude that t_2 must have fired for a total of $q' \in \mathbb{N}$ times whenever we observe event c for q' times. As a result, the information of events c and d can be used to "estimate" the actual number of firings of t_1 and t_2. This motivates us to use such structural information to obtain a monitor-based supervisor.

In the following we introduce the notion of *transition dependency* that could be used to better estimate the firings of these transitions, after the introduction of *private subnets*.

Definition 6. (Private Subnet). Given a pair (T', T'') where $T', T'' \subseteq T$, the (T', T'')*-private subnet* is the subnet of N obtained by removing all places $p \in P$ such that for all $(t', t'') \in T' \times T''$, there does not exist a simple path π from t' to t'' such that $p \in \pi$, followed by removing all isolated transitions. □

An example of the private subnet is given in Example 3. Note that in general an (X, Y)-private subnet is different from the (Y, X)-private subnet. The dependency of transitions, which is a structural property, is based on the private subnet and will be used in the next section to establish a quantitative relation between the firing of transitions in T' and in T''.

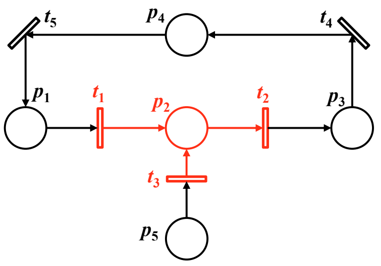

Fig. 2. A Petri net for Example 3.

Definition 7. Given a Petri net N and two sets of transitions $T', T'' \subseteq T$, let $\hat{N}$ be the (T', T'')-private subnet. Transition set T'' is said to be *dependent* on T' if for any marking $\hat{M}$ in $\hat{N}$ there exists an integer $K \in \mathbb{N}$ such that for any firing sequence $\sigma \in L(\hat{N}, \hat{M})$ such that $\sigma(T') = 0$, $\sum_{t \in T''} y_\sigma(t) \leq K$ holds. □

By Definition 7, if T'' is dependent on T', then in the private subnet of (T', T''), transitions in T'' cannot fire infinitely often without firing transitions in T', regardless the initial marking. In such a case, the firings of transitions in T'' can be used to estimate the firings of transitions in T'.

It is worth to note that a similar notion, called the *synchronic distance*, was proposed in (Silva and Colom, 1988). Roughly speaking, the synchronic distance between sets T' and T'' is the maximal times that transitions in one set can fire without any firing of transitions in the other. However, it is different from Definition 7 since the synchronic relation in (Silva and Colom, 1988) applies for the whole net N instead of the private subnet, and hence the existence of a finite synchronic distance between T' and T'' does not necessarily imply the dependency. Moreover, the synchrony theory is more tailored to characterize *fairness* in Petri nets and cannot be directly used for the purpose of control in LPNs.

Example 3. Consider the net in Figure 2. Let $T_1 = \{t_1\}$ and $T_2 = \{t_2\}$. By Silva and Colom (1988) the synchronic distance from T_1 to T_2 is $r = M_0(p_2) + M_0(p_5)$ that is always finite (i.e., t_2 can fire at most r times without firing t_1). However, consider the (T_1, T_2)-private subnet is the red part of the net, consisting of t_1, t_2, t_3, and p_2. Clearly, due to transition t_3, transition t_2 can fire infinite number of times without firing t_1, and hence T_2 is not dependent on T_1. □

The following proposition provides us a way to determine if T'' is dependent on T'.

Proposition 1. Given a net N and two sets of transitions $T', T'' \subseteq T$, T'' is dependent on T' if the following system of constraints is feasible.

$$\begin{cases} C_{\hat{N}} \cdot \mathbf{y} \geq \mathbf{0} \\ \mathbf{y} \geq \mathbf{0} \\ \sum\limits_{t' \in T'} y(t') = 0 \\ \sum\limits_{t'' \in T''} y(t'') \geq 1 \end{cases} \tag{2}$$

where $\hat{N}$ is the (T', T'')-private subnet.

Proof: If Eq. (2) admits at least one solution $\mathbf{y} \gneq \mathbf{0}$, it means that $\mathbf{y}$ contains some transitions in T'' but no transition in T'. This indicates that there necessarily exists a marking $\hat{M}$ such that in the net $\langle \hat{N}, \hat{M} \rangle$ from which a firing sequence σ associated to $\mathbf{y}$ can repeatedly fire, which leads to arbitrary large number of firings of transitions in T'' without firing any of T'. Hence T'' is not dependent on T'.

On the other hand, if T'' is not dependent on T', from some marking M there exists a repetitive firing sequence that contains transitions in T'' but no transition in T'. The firing vector of this sequence is an admissible solution of Eq. (2). □

5. FEEDBACK CONTROL POLICY DESIGN

As discussed in Section III, the conservativeness of a supervisor comes from the fact that the firing of an input/output transition of the GMEC cannot be identified from its spurious ones. By taking advantage of transition dependency the firing of spurious transitions can be identified. However, the condition in Definition 7 does not provide a quantitative relationship between the firing of T' and T'', which will be solved in this section.

5.1 Identifying the Firing of an Input Transition

Suppose that a transition t labelled $\hat{e}$ is an input transition of $(\mathbf{w},k)$. Since the supervisor has no information about the current marking, each time it observes $\hat{e}$ it cannot determine if t or one of its spurious transactions in the set $S(t)$ has fired. Thus to be on the safe side, it has to assume that this observation $\hat{e}$ is due to the firing of t. However, to detect the firings of spurious transitions $S(t)$, we can sometimes monitor the occurrence of another event $e \neq \hat{e}$ such that T_e is dependent on $S(t)$.

Suppose that by solving the ILPP in Proposition 1 we can find an event e such that the set T_e is dependent on $S(t)$. Such dependency guarantees that by observing event e for sufficiently many times we can conclude that some transitions in $S(t)$ must have fired. Hence in the following we aim to find a function $\alpha : \mathbb{N} \to \mathbb{N}$ such that by observing event e for q times we can conclude that the transitions in $S(t)$ must have fire at least $\alpha(q)$ times.

Note that since there are some places and transitions that do not affect the relationship between q and $\alpha(q)$ in any cases, it is sufficient to consider the $(S(t),T_e)$*-private subnet* instead of the whole net.

In the following we show that if an $(S(t),T_e)$-private subnet is zero-marked, i.e., its initial marking M_0 is $\mathbf{0}$, the function α can be obtained thanks to the *Hilbert Basis*. In plain words, the Hilbert Basis of a convex cone C is a minimal set of integer vectors such that every integer vector in C is a conical combination of the vectors in the Hilbert Basis with nonnegative integer coefficients.

Definition 8. (Hilbert Basis). Let J be the set of all integer solutions of a system of linear inequalities $\mathbf{A}\cdot\mathbf{x} \geq \mathbf{0}$. A set $H \subset J$ is called a *basis of solutions* of $\mathbf{A}\cdot\mathbf{x} \geq \mathbf{0}$ if H is a minimal set such that every element in J is a linear combination of vectors in H with nonnegative integer coefficients. In particular, H is called the *Hilbert basis* of J if J is nonnegative. □

The Hilbert basis is unique. Algorithms to compute it have been presented in the literature (Chubarov and Voronkov, 2005) and several softwares such as *Normaliz* (Bruns and Ichim, 2010) have been developed. Now we can state the following lemma which provides a quantitative relationship between two dependent transition sets.

Lemma 1. Given two sets of transitions T' and T'' such that T'' is depended on T', for a zero-marked subnet $\langle N',\mathbf{0}\rangle$ where N' is the (T',T'')-private subnet, the following condition holds:

$$\max_{\sigma\in L(N',\mathbf{0})}\{\sum_{\hat{t}\in T''} y_\sigma(\hat{t})/\sum_{\hat{t}\in T'} y_\sigma(\hat{t})\} \leq \max_{y\in H}\{\sum_{\hat{t}\in T''} y(\hat{t})/\sum_{\hat{t}\in T'} y(\hat{t})\}$$

where H is the set of Hilbert basis of $C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$.

Proof: Since the net is zero-marked and T'' is dependent on T', the maximal value of $\sum_{\hat{t}\in T''} y(\hat{t})/\sum_{\hat{t}\in T'} y(\hat{t})$ for $y \in H$ on the right-hand side is always finite (i.e., the denominator must be non-zero).

Since the net is initially zero-marked, for any $\sigma \in L(N',\mathbf{0})$, $C\cdot\mathbf{y}_\sigma \geq \mathbf{0}$ holds. Let J be the set $J = \{\mathbf{y} \in \mathbb{N}^{|T'|} \mid C\cdot\mathbf{y} \geq \mathbf{0}\}$ and H be its Hilbert basis. For any $\mathbf{y} \in J$ there exists a vector $\mathbf{x} \in \mathbb{N}^{|H|}$ such that $\mathbf{y} = \sum_{i=1}^{|H|} x(i)\cdot\mathbf{y}_i$ where $\mathbf{y}_i \in H$ is the i-th component of the Hilbert basis. Hence we have:

$$\begin{cases} \dfrac{\sum_{\hat{t}\in T''} y(\hat{t})}{\sum_{\hat{t}\in T')} y(\hat{t})} & = \dfrac{\sum_{i=1}^{|H|} x(i)\cdot\sum_{\hat{t}\in T''} y_i(\hat{t})}{\sum_{i=1}^{|H|} x(i)\cdot\sum_{\hat{t}\in T'} y_i(\hat{t})} \\ & \leq \max\limits_{\mathbf{y}\in H} \dfrac{\sum_{\hat{t}\in T''} y(\hat{t})}{\sum_{\hat{t}\in T'} y(\hat{t})}. \end{cases}$$

□

By Lemma 1 we immediately have the following result to establish a relation between the firing of T_e and $S(t)$.

Theorem 1. Given a net $\langle N',\mathbf{0}\rangle$ where N' is the $(S(t),T_e)$-private subnet, the following condition holds:

$$\max_{\sigma\in L(N',\mathbf{0})}\{\sum_{\hat{t}\in T_e} y_\sigma(\hat{t})/\sum_{\hat{t}\in S(t)} y_\sigma(\hat{t})\} \leq \max_{y\in H}\{\sum_{\hat{t}\in T_e} y(\hat{t})/\sum_{\hat{t}\in S(t)} y(\hat{t})\}$$

where H is the set of Hilbert basis of $C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$.

Proof: Straightforwardly from Lemma 1 by letting $T' = S(t)$ and $T'' = T_e$. □

Corollary 1. Given a net $\langle N',\mathbf{0}\rangle$ where N' is the $(S(t),T_e)$-private subnet, for any $\sigma \in L(N',\mathbf{0})$ it holds:

$$(\sigma(T_e) = q) \Rightarrow (\sigma(S(t)) \geq \lceil q/r \rceil)$$

where $r = \max_{\mathbf{y}\in H} \frac{\sum_{\hat{t}\in T_e} y(\hat{t})}{\sum_{\hat{t}\in S(t)} y(\hat{t})}$, and "$\lceil\cdot\rceil$" is the *ceiling operator* which returns the minimal integer that is not smaller than $(\cdot)$.

Proof: Trivial, otherwise $\mathbf{y}_\sigma$ is a solution of $C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$ but cannot be obtained by nonnegative integer linear combination of elements in H, a contradiction. □

It is worth noting that if the subnet N' is not zero-marked, Lemma 1 holds by replacing the inequality $C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$ by $M_0 + C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$, and hence the results in Theorem 1 and Corollary 1 are still applicable in these cases. However, the resulted control policy will be more conservative than necessary, since the firing vector in $\mathbf{y} \in H$ with the maximal value of $\sum_{t\in T_e} y(t)/\sum_{t\in S(t)} y(t)$ may not satisfy $C_{N'}\cdot\mathbf{y} \geq \mathbf{0}$. Such situation requires to be handled with a special care and this is an issue that will be explored in our future work. However, we point out that in many Petri net models of real systems tokens are initially distributed only in some *idle* places while the working zone (where those input/output/spurious transitions are associated to) is zero-marked. Theorem 1 and Corollary 1 (and results in the rest of this section) are well applicable to these cases.

5.2 Identifying the Firing of an Output Transition

The case to correct the firing of output transition is similar. Suppose that a transition t labelled $\hat{e}$ is an output transition of $(\mathbf{w},k)$. Since the supervisor has no information about the current marking, each time it observes $\hat{e}$ it cannot determine if t or one of its spurious transitions in the set $S(t)$ (assuming that it is not empty) has fired. Thus to be on the safe side, it has to assume that this observation $\hat{e}$ is due to the firing of a transition in $S(t)$. So to detect the firings of the *real* output transition t we use another event $e \neq \hat{e}$ such that T_e is dependent on t. Analogous to Theorem 1 we have the following result on the quantitative function $\beta : \mathbb{N} \to \mathbb{N}$ between the occurrence of e and the firing of t. Their proofs are analogous to that of Theorem 1 and Corollary 1 and hence are omitted.

Theorem 2. Given a net $\langle N', \mathbf{0}\rangle$ where N' is the (T_t, T_e)-private subnet, the following condition holds:

$$\max_{\sigma \in L(N',\mathbf{0})} \{(\sum_{\hat{t} \in T_e} y_\sigma(\hat{t}))/y_\sigma(t)\} \leq \max_{\mathbf{y} \in H} \{(\sum_{\hat{t} \in T_e} y(\hat{t}))/y(t)\}$$

where H is the set of Hilbert basis of $C' \cdot \mathbf{y} \geq \mathbf{0}$.

Corollary 2. Given a net $\langle N', \mathbf{0}\rangle$ where N' is the (T_t, T_e)-private subnet, the following condition holds:

$$(\sigma(T_e) = q) \Rightarrow (\sigma(t) \geq \lceil q/r \rceil)$$

where $r = \max_{\mathbf{y} \in H} \frac{\sum_{\hat{t} \in T_e} y(\hat{t})}{y(t)}$.

To conclude the previous two subsections we have the following remarks. The objective of our approach is to improve the estimate of the number of firings of input/output transitions of a given GMEC, based on identifying the firings of spurious transitions. However, such estimate is not guaranteed to converge to the *real* number of firings. Hence the control behavior may be overconservative with respect to the marking-estimation-based approaches in some cases. However, our method has a low online computational load, and it is robust with respect to the change of the net structure. To perform a better estimation is part of our future work.

5.3 Online Feedback Control Policy

In the following we propose an online control policy based on the previously presented results.

Algorithm 1 consists of two stages. In the offline stage, it first removes all initially marked places so that the remaining part of the net is zero-marked. Then Steps 2 to 10 compute all input and all output transitions and find out those events that are dependent on $S(t)$ (if t is an input transition) or t (if t is an output transition), according to Proposition 1. To simplify the presentation we denote such relation as a partial function $D : T \to E$, i.e., $D(t) = e$ if t is an input (resp. output) transition of $(\mathbf{w}, k)$ and e is the event selected during the offline stage, which is dependent on $S(t)$ (resp. t).

During the online stage, in each iteration from Steps 12 to 18 the supervisor makes a control decision by preventing transitions whose occurrence may violate the control demand ($\mathbf{w}^T \cdot M > k$). From Steps 19 to 27, once an event e is observed, the supervisor records this e if one of the following three cases is verified: (1) there exists an input transition t_{in} whose label is e; (2) the set T_e is dependent on the spurious transitions $S(t_{in})$ of an input transition t_{in}; (3) the set T_e is dependent on an output transition t_{out}. Then Step 26 is used to compute the maximal token count so far. The following theorem guarantees the correctness of Step 26.

Theorem 3. Given a net $\langle N, M_0\rangle$ and a GMEC satisfying Assumption 1, for an observation $Obs \in E^*$, the token count of $(\mathbf{w}, k)$ at marking M reached from M_0 by firing a sequence σ such that $\ell(\sigma) = Obs$ satisfies:

$$\begin{aligned} \mathbf{w}^T \cdot M \leq\ & \mathbf{w}^T \cdot M_0 \\ & + \sum_{t \in I_\mathbf{w}} \mathbf{w}^T \cdot C(\cdot,t) \cdot (Rec(\ell(t)) - \alpha_t(Rec(D(t)))) \\ & + \sum_{t \in O_\mathbf{w}} \mathbf{w}^T \cdot C(\cdot,t) \cdot \beta_t(Rec(D(t))) \end{aligned} \quad (3)$$

Proof: The first line corresponds to the initial token count. In the second line, for each input transition t, $Rec(\ell(t))$ records the occurrence of event $e = \ell(t)$ generated by either t itself or its spurious transitions $S(t)$, while according to Theorem 1 and Corollary 1 $\alpha_t(Rec(D(t)))$ indicates at least how many $e = \ell(t)$ is not generated by t but by its spurious transitions. In the third line, for each output transition t, while according to Theorem 2 and Corollary 2 $\beta_t(Rec(D(t)))$ indicates at least how many $e = \ell(t)$ is guaranteed to be generated by t. By taking all these into account the inequality holds. □

Algorithm 1 Supervisor Design

```
Input: A labelled Petri net $G = (N, M_0, E, \ell)$ and a GMEC $(\mathbf{w}, k)$ satisfying Assumption 1.
Offline Stage:
1: Remove all places from $N$ such that $M_0(p) > 0$;
2: Let $I_\mathbf{w} = \{t \mid \mathbf{w}^T \cdot C(\cdot,t) > 0\}$, let $O_\mathbf{w} = \{t \mid \mathbf{w}^T \cdot C(\cdot,t) < 0\}$;
3: for all $t \in I_\mathbf{w}$, do
4:     Find event $e$ such that $T_e$ is dependent on $S(t)$, let $D(t) = e$;
5:     Compute $\alpha_t : \mathbb{N} \to \mathbb{N}$ according to Corollary 1;
6: end for
7: for all $t \in O_\mathbf{w}$, do
8:     Find event $e$ such that $T_e$ is dependent on $t$, let $D(t) = e$;
9:     Compute $\beta_t : \mathbb{N} \to \mathbb{N}$ according to Corollary 2;
10: end for
Online Stage:
11: Let $Rec = \mathbf{0}$, $Sum = \mathbf{w}^T \cdot M_0$;
12: Let $Ctrl = E$;
13: for all $t \in I_\mathbf{w}$, do
14:     if $Sum + \mathbf{w}^T \cdot C(\cdot,t) > k$, then
15:         $Ctrl = Ctrl \setminus \{\ell(t)\}$;
16:     end if
17: end for
18: Execute $Ctrl$;
19: if an event $e$ is observed, then
20:     if $\exists t_{in} \in I_\mathbf{w} \cap Ctrl : \ell(t_{in}) = e$, then
21:         $Rec(e) = Rec(e) + 1$;
22:     end if
23:     if $\exists t_{in} \in I_\mathbf{w}$ and $T_e$ is dependent on $S(t_{in})$, then
24:         $Rec(e) = Rec(e) + 1$;
25:     end if
26:     if $\exists t_{out} \in O_\mathbf{w}$ and $T_e$ is dependent on $t_{out}$, then
27:         $Rec(e) = Rec(e) + 1$;
28:     end if
29:     let $Sum = \mathbf{w}^T \cdot M_0 + \sum_{t \in I}(\mathbf{w}^T \cdot C(\cdot,t) \cdot (Rec(\ell(t)) - \alpha_t(Rec(D(t))))) + \sum_{t \in O}(\mathbf{w}^T \cdot C(\cdot,t) \cdot \beta_t(Rec(D(t))))$;
30: end if
31: Goto Step 12;
```

6. EXAMPLE

Consider the LPN in Figure 3 and a GMEC $(\mathbf{w}, k)$ to be enforced, representing markings satisfying $M(p_2) + M(p_3) \leq 3$. Clearly $(\mathbf{w}, k)$ is observable but not distinguishable. Now we use the proposed method to design a monitor-based supervisor that enforces $(\mathbf{w}, k)$.

Since t_1 labelled a is an input transition of $(\mathbf{w}, k)$, we need to find an event dependent on $S(t_1) = \{t_6, t_9\}$ to distinguish $S(t_1)$ from t_1. By solving ILPP (2) we understand that event d is not suitable since event d can occur infinitely often without any firing of $S(t_1)$. On the other hand, event c can be used to indicate the firing of $S(t_1)$. The private subnet of $(\{t_6, t_9\}, \{t_7, t_{12}\})$, denoted as N', is marked in red. The Hilbert basis H of $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$ contains 36 firing vectors, among which vector $\mathbf{y}^*$ representing $2 \cdot t_9 + 6 \cdot t_{10} + 6 \cdot t_{11} + 3 \cdot t_{12}$ has the maximal quantity of $[y(t_7) + y(t_{12})]/[y(t_6) + y(t_9)] = 3/2$. By Theorem 1 and Corollary 1

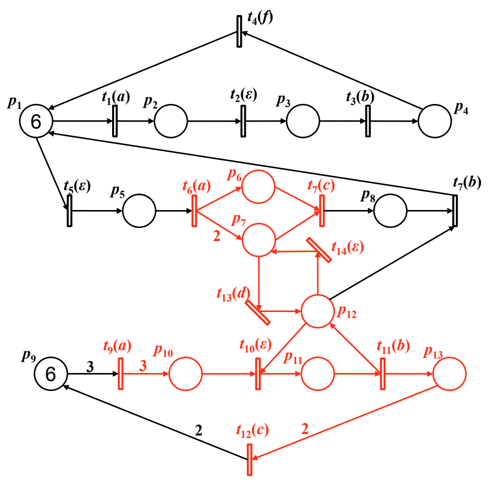

Fig. 3. The example for Section VI.

whenever event c is observed q times, we can guarantee that at least $\lceil 2/3 \cdot q \rceil$ of the occurrence of event a is generated by t_6 or t_9 instead of t_1.

On the other hand, transition t_3 is an output transition of $(\mathbf{w}, k)$ and its firing can be identified by event f (by t_4) in a 1:1 ratio. Hence the online supervisor is such that:

(1) Set $Sum = 0$, $Rec(a) = Rec(c) = Rec(f) = 0$;
(2) If $Sum + \mathbf{w}^T \cdot C(\cdot, t_1) = Sum + 1 > 3$, i.e., $Sum = 3$, disable transition t_1;
(3) Wait until an event $e \in \{a, c, f\}$ occurs and update $Rec(e) = Rec(e) + 1$;
(4) Update Sum by using Eq. (3):
$$Sum = Rec(a) - \lceil 2/3 \cdot Rec(c) \rceil - Rec(f)$$
and goto Step 2;

This supervisor only monitors events a, c, and f online instead of all events. One can readily verify that such supervisor is robust to all initial marking $M_0 = r \cdot p_1 + r \cdot p_9$ with different r. Moreover, by some modification on the structure that is irrelevant to events a, b, c, f (e.g., by letting $\ell(t_{10}) = d$ and $\ell(t_{13}) = \varepsilon$), our supervisor can also be used without any modification.

7. CONCLUSION

In this paper we propose an online feedback control method to enforce a GMEC in labelled Petri nets that contains unobservable and indistinguishable transitions. We present an algorithm to design an online supervisor based on the computation of *Hilbert basis* which solely depends on the net structure and has low online computational cost. Such a supervisor is robust with respect to the change of the initial markings and of the irrelevant part of the plant net.

REFERENCES

Basile, F., Cordone, R., and Piroddi, L. (2013). Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models. *Automatica*, 49(11), 3432–3439.

Basile, F., Cordone, R., and Piroddi, L. (2015). A branch and bound approach for the design of decentralized supervisors in Petri net models. *Automatica*, 52, 322–333.

Bruns, W. and Ichim, B. (2010). Normaliz: Algorithms for affine monoids and rational cones. *Journal of Algebra*, 324(5), 1098–1113.

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

Cabasino, M.P., Giua, A., Hadjicostis, C.N., and Seatzu, C. (2015). Fault model identification and synthesis in Petri nets. *Discrete Event Dynamic Systems*, 25(13), 419–440.

Chubarov, D. and Voronkov, A. (2005). Basis of solutions for a system of linear inequalities in integers: Computation and applications. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, 260–270. Springer Berlin Heidelberg.

Giua, A., DiCesare, F., and Silva, M. (1992). Generalized mutual exclusion constraints for Petri nets with uncontrollable transitions. In *In Proceedings of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, 947–949. Chicago, USA.

Li, Z.W., Wu, N.Q., and Zhou, M.C. (2012). Deadlock control of automated manufacturing systems based on Petri nets – a literature review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(4), 437–462.

Li, Z.W. and Zhou, M.C. (2004). Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(1), 38–51.

Luo, J.L., Wu, W.M., Su, H.Y., and Chu, J. (2009). Supervisor synthesis for enforcing a class of generalized mutual exclusion constraints on Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 39(6), 1237–1246.

Luo, J. and Zhou, M. (2017). Petri-net controller synthesis for partially controllable and observable discrete event systems. *IEEE Transactions on Automatic Control*, 62, 1301–1313.

Ma, Z.Y., Li, Z.W., and Giua, A. (2015). Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints. *IEEE Transactions on Automatic Control*, 60, 1774–1785.

Ma, Z.Y., Li, Z.W., and Giua, A. (2017a). Characterization of admissible marking sets in Petri nets with conflicts and synchronizations. *IEEE Transactions on Automatic Control*, 62(3), 1329–1341.

Ma, Z.Y., Tong, Y., Li, Z.W., and Giua, A. (2017b). Basis marking representation of Petri net reachability spaces and its application to the reachability problem. *IEEE Transactions on Automatic Control*, 62(3), 1078–1093.

Moody, J. and Antsaklis, P. (2000). Petri net supervisors for DES with uncontrollable and unobservable transitions. *IEEE Transactions on Automatic Control*, 45(3), 462–476.

Nazeem, A. and Reveliotis, S. (2015). Maximally permissive deadlock avoidance for resource allocation systems with r/w-locks. *Discrete Event Dynamic Systems*, 25(1), 31–63.

Ramadge, P.J. and Wonham, W.M. (1989). The control of discrete event systems. *Proceedings of IEEE*, 77(1), 81–98.

Ru, Y., Cabasino, M.P., Giua, A., and Hadjicostis, C.N. (2014). Supervisor synthesis for discrete event systems under partial observation and arbitrary forbidden state specifications. *Discrete Event Dynamic Systems*, 24(3), 275–307.

Silva, M. and Colom, J.M. (1988). *On the computation of structural synchronic invariants in P/T nets*, 386–417. Springer Berlin Heidelberg.