State estimation and control of nondeterministic λ-free labeled Petri nets*

Alessandro Giua, Daniele Corona, Carla Seatzu

Dip. Ingegneria Elettrica ed Elettronica, Università di Cagliari,
Piazza d'Armi, 09123 Cagliari, Italy.
Email: {giua,daniele.corona,seatzu}@diee.unica.it

Abstract

In this paper we present an original approach to estimate the marking of a labeled Petri net based on the observation of transition labels. In particular, we consider the case of nondeterministic transitions, i.e., transitions that share the same label and that can be simultaneously enabled. We also show how the estimate generated by the observer may be used to design a state feedback controller for forbidden marking specifications. More precisely, we discuss two different cases: the label-based feedback and the transition-based feedback, depending on the possibility of the controller to distinguish among transitions with the same label.

1 Introduction

This paper deals with the problem of estimating the marking of a nondeterministic λ-free labeled Petri net based on the observation of transition labels. We also show how an observer constructed following the proposed approach can be used in a state-feedback control loop.

The problem of estimating the state of a dynamic system is a fundamental issue in *system theory*. A similar problem has also been addressed in theoretical *computer science* within the framework of nondeterministic language generators. Nevertheless, the problem statement is quite different depending on the considered framework.

In system theory, a state observer reconstructs the plant states that cannot be measured on the basis of the observation of some physical variables. The initial state of the system is completely unknown, while a perfect knowledge of the system dynamics is usually assumed, i.e., the behaviour of the system is *deterministic*.

Analogous problems in the case of discrete event systems (DES) have been discussed in the literature for systems represented as finite automata [1, 2, 7, 10, 12] or Petri nets [5].

In the context of computer science, on the contrary, the system may be nondeterministic, i.e., partially known, and the observed word of events does not contain all information of the sequence of transitions that have been executed. Thus, the problem of observation consists in reconstructing the system state on the basis of the observation of the words of events.

In this paper we explore the possibility of using Petri nets as discrete event models and address the observer design from a computer science point of view. We consider nondeterministic λ-free Petri nets

*Published as: A. Giua, D. Corona, C. Seatzu, "State estimation and control of nondeterministic λ-free labeled Petri nets," *Proc. IFAC WODES04: 7th Workshop on Discrete Event Systems* (Reims, France), Sep 2004.

where the only cause of nondeterminism arises because two or more transitions with the same label may be simultaneously enabled from a reachable marking.

We propose an approach to build a state observer that does not require the construction of the reachability graph, and thus works for both bounded and unbounded PN. Under some assumptions, we show that the set of markings that are consistent with the observation w — denoted $\mathcal{C}(w)$ — can be written as the solution of a linear system with a fixed structure that depends on some parameters that can be recursively computed. The main advantage of the proposed approach is that we need not exhaustively enumerate all consistent markings.

Other authors [8] have also discussed the problem of estimating the marking of a Petri net using a mix of transition firings and place observations. Finally, Zhang and Holloway [13] used a Controlled Petri Net model for forbidden state avoidance under partial *event* observation with the assumption that the initial marking be known.

The characterization used to define the set of consistent markings — as well as the algorithm that is used to implement it — has been firstly proposed by the authors in [3] as a conjecture. In this paper we present a slightly modified version of this algorithm (see Algorithm 6) for which a formal proof has been obtained: the complete proof is not reported here for lack of space but can be found in [6].

In this paper we also show how the proposed linear algebraic characterization of the set of consistent markings can be used to design a feedback controller for forbidden marking specifications. In particular, we assume that specifications are given in the form of Generalized Mutual Exclusion Constraints (GMEC) that limit the weighted sum of tokens in subsets of places.

We discuss two different control laws.

Label-based feedback. In this case the controller cannot distinguish among transitions that are undistinguishable by the observer (namely, transitions with the same label). Hence, if a transition labeled by the event e is disabled, then the control pattern must simultaneously disable all other transitions labeled by e.

Transition-based feedback. In this case we assume that the controller is not constrained by the observer observation mask, and may assign a different control pattern to different transitions even if they share the same label e.

It is important to observe that in the case of label-based feedback, Algorithm 6 can still be used to compute the set of markings in which the *closed-loop system* may be, given the actual observation.

On the contrary, in the case of transition-based feedback the control pattern may reduce the nondeterminism of the net: when the nondeterministic event e is observed, only a subset of transitions labeled e may have fired, namely those transitions that are control enabled. This implies that Algorithm 6 needs to be slightly modified when used to compute the set of markings in which the *closed-loop system* may be, given the actual observation. This point is formalized in Algorithm 10.

An original example of state estimation and control, taken from the manufacturing domain, that motivates the interest for the particular considered model, is finally presented in Section 6.

2 Background on Petri nets

In this section we recall the formalism used in the paper. For more details on Petri nets we address to [9].

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set of n transitions; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre–* and *post–* incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix. The *preset* and *postset* of a node $X \in P \cup T$ are denoted ${}^\bullet X$ and $X^\bullet$ while ${}^\bullet X^\bullet = {}^\bullet X \cup X^\bullet$.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number

of tokens, represented by black dots. We denote $M(p)$ 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.

We write $M\ [\varsigma\rangle\ M'$ to denote that the firing of ς yields M'. We also denote $\vec{\sigma}\ :\ T \to \mathbb{N}$ the *firing vector* associated to a sequence ς.

The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0\rangle$ and is denoted $R(N, M_0)$.

A *labeling function* $L : T \to E$ assigns to each transition $t \in T$ a symbol from a given alphabet E. Note that the same label $e \in E$ may be associated to more than one transition while no transition may be labeled with the empty string ε. Using the notation of [11] and [4] we say that this labeling function is λ*-free*.

Definition 1. *A Petri net system $\langle N, M_0\rangle$ with λ-free labeling function $L : T \to E$ is* deterministic *if for all markings $M \in R(N, M_0)$ and for any two transitions $t, t' \in T$:*

$$t \neq t',\ L(t) = L(t'),\ M[t\rangle \implies \neg M[t'\rangle,$$

i.e., if two transitions are labeled with the same symbol they cannot simultaneously be enabled at M. ■

Determinism is a behavioral property but it is also possible to introduce a structural definition of determinism.

Definition 2. *A Petri net N with λ-free labeling function $L : T \to E$ is* structurally deterministic *if for any two transitions $t, t' \in T$:*

$$t \neq t' \implies L(t) \neq L(t'),$$

i.e., two different transitions cannot be labeled with the same symbol. ■

Note that if a Petri net N is structurally deterministic, then the net system $\langle N, M_0\rangle$ is deterministic for all initial marking M_0.

In this paper we consider Petri nets that are not structurally deterministic. We say that a transition t is *nondeterministic* if its label is also associated to other transitions, otherwise a transition t is said to be *deterministic*. We also denote T^d the set of deterministic transitions and T^n the set of nondeterministic transitions. Clearly, $T = T^d \cup T^n$.

Analogously, we say that an event e is nondeterministic if there exists more than one transition t such that $L(t) = e$, otherwise we say that the event e is deterministic. Therefore, with no ambiguity on the notation, we may write $E = E^d \cup E^n$.

Note that the labeling function restricted to T^d is an isomorphism and thus, with no loss of generality we can assume $E^d = T^d$.

We denote as T_e the set of transitions labeled e, i.e,

$$T_e = \{t \in T\ \mid\ L(t) = e\}.$$

The restriction of the incidence matrix C to T_e (T^n) is denoted C_e (C^n) and the restriction of the firing vector $\vec{\sigma}$ to T_e is denoted $\vec{\sigma}_e$.

Finally, to each set of nondeterministic transitions T_e we associate the set $\mathcal{T}_e$ containing all possible subsets of transitions, apart from itself and the empty set, i.e.,

$$\mathcal{T}_e = \{\tau \subseteq T_e\ \mid\ \tau \neq \emptyset,\ \tau \neq T_e\} = 2^{T_e} \setminus \{\emptyset, T_e\}.$$

Clearly, $|\mathcal{T}_e| = 2^{n_e} - 2$ where n_e denotes the number of nondeterministic transitions labeled e.

We denote as w the word of events associated to the sequence ς, i.e., $w = L(\varsigma)$.

3 A linear algebraic characterization of the set of consistent markings

In this paper we deal with the problem of estimating the marking of a net system $\langle N, M_0\rangle$ whose marking cannot be directly observed.

After the word w has been observed, we define the set $\mathcal{C}(w)$ of w-consistent markings as the set of all markings in which the system may be given the observed behavior.

Definition 3. *Given an observed word w, the set of* w-consistent markings *is* $\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \exists$ *a sequence of transitions* $\varsigma : M_0[\varsigma\rangle M$ *and* $L(\varsigma) = w\}$. ■

In [6] we provided a systematic and efficient procedure to estimate the set of markings that are consistent with an observed word under the following assumptions:

(A1) The structure of the net N is known.

(A2) The initial marking M_0 is known.

(A3) The label function is λ-free and labels associated to transition firings can be observed.

(A4) Nondeterministic transitions are contact free, i.e., for any two nondeterministic transitions t_i and t_j, it holds that ${}^\bullet t_i^\bullet \cap {}^\bullet t_j^\bullet = \emptyset$.

Clearly, $\mathcal{C}(\varepsilon) = M_0$ and $\mathcal{C}(w)$ is a singleton if for all e in w, T_e is a singleton. On the contrary, the degree of nondeterminism may increase as the cardinality of T_e increases.

In [6] we have shown that, under the assumptions (A1) to (A4), a fixed number of constraints, not depending on the length of the observed word w, may be used to describe the set of w-consistent markings.

Let us first introduce the following notation.

Definition 4. *Given a marking M and a transition $t \in T$, we define*

$$z(M,t) = \min_{p \in {}^\bullet t} \left\{ \left\lfloor \frac{M(p)}{Pre(p,t)} \right\rfloor \right\}$$

the enabling degree of transition t at M.

Given a set of transitions $\tau \subseteq T$, we also define

$$z(M,\tau) = \sum_{t \in \tau} z(M,t).$$

Finally, given a vector $\vec{\sigma} \in \mathbb{N}^n$, we denote as

$$\sigma(\tau) = \sum_{t \in \tau} \sigma(t).$$

■

Note that if all transitions in τ are conflict free, then $z(\tau)$ represents the number of times transitions in τ may simultaneously fire at M.

Theorem 5 ([6])**.** *Let us consider a labeled Petri net system $\langle N, M_0\rangle$ and let $L : T \to E$ be its labeling function. Let assumptions (A1) to (A4) be verified. Then, for all words $w \in E^*$ the set of w-consistent markings $\mathcal{C}(w)$ is equal to*

$$\begin{aligned} \mathcal{C}(w) = \mathcal{M}(w) \stackrel{\text{def}}{=} \{M \in \mathbb{N}^m \mid \\ M = M_{b,w} + \sum_{e \in E^n} C_e \vec{\sigma}_e; \quad \vec{\sigma}_e \in \mathcal{S}_e(w)\} \end{aligned} \tag{1}$$

where

$$\begin{aligned} \mathcal{S}_e(w) \stackrel{\text{def}}{=} \{\vec{\sigma} \in \mathbb{N}^{n_e} \;\mid\; (\forall \tau \in \mathcal{T}_e) \quad & \sigma(\tau) \le u_w(\tau), \\ & \sigma(T_e) = u_w(T_e)\}, \end{aligned} \tag{2}$$

and the upper bounds $u_w(\tau)$ and $u_w(T_e)$, as well as the marking $M_{b,w}$, are computed using the recursive Algorithm 6.

Therefore, the number of constraints used to describe the set $\mathcal{S}_e(w)$ is equal to $2^{n_e} - 1$, regardless of the length of the observed word w.

Algorithm 6 (Upper bounds and basis marking computation)**.**

1. Let $w = \varepsilon$ and $M_{b,w} = M_0$.
2. Let $u_w(\tau) = 0$ for all $e \in E^n$ and for all $\tau \in \mathcal{T}_e$.
3. Let $u_w(T_e) = 0$ for all $e \in E^n$.
4. Wait until an event e is observed.
5. Let $flag = 0$.
6. If $e \in E^d$, then
  let $t = L^{-1}(e)$,
  if ${}^\bullet t \cap ({}^\bullet T^n {}^\bullet) = \emptyset$, then (Case A)
    $M_{b,we} = M_{b,w} + C(\cdot, t)$
  endif
  if $P_t \stackrel{\text{def}}{=} {}^\bullet t \cap (T^n {}^\bullet) \neq \emptyset$, then (Case B)
    $\vec{\sigma}_\alpha = \vec{0}$ (a vector of dimension $|T^n| \times 1$)
    $flag = 1$
    $T_{up} \stackrel{\text{def}}{=} T^n \cap^\bullet P_t$
    for all $\hat{t} \in T_{up}$, then
      let $\sigma_\alpha(\hat{t}) = \max\limits_{p \in P_t} \left\{ 0, \left\lceil \dfrac{Pre(p,t) - M_{b,w}(p)}{Post(p,\hat{t})} \right\rceil \right\}$
    endfor
    for all $\tau \ : \ \tau \cap T_{up} \neq \emptyset$, then
      $u_{we}(\tau) = u_w(\tau) - \sum\limits_{t \in \tau} \sigma_\alpha(t)$
    endfor
    $M_{b,we} = M_{b,w} + C(\cdot, t) + C^n \vec{\sigma}_\alpha$
  endif
  if ${}^\bullet t \cap ({}^\bullet T^n) \neq \emptyset$, then (Case C)
    if $flag = 0$, then
      $M_{b,we} = M_{b,w} + C(\cdot, t)$
    endif
    let $\mathcal{T}_r(t) = \{\hat{t} \in T^n \ \mid \ {}^\bullet t \cap^\bullet \hat{t} \neq \emptyset\}$
    for all $\hat{t} \in \mathcal{T}_r(t)$, then
      $u_{we}(\{\hat{t}\}) = \min\{u_w(\{\hat{t}\}), z(M_{b,we}, \hat{t})\}$
      for all $\tau \in \mathcal{T}_{L(\hat{t})} : \hat{t} \in \tau$ with $\tau \neq \{\hat{t}\}$, then
        $u_{we}(\tau) = \min\{u_w(\tau), u_{we}(\{\hat{t}\}) + u_w(\tau \setminus \{\hat{t}\})\}$
      endfor
    endfor
  endif
else (Case D)
  for all $\tau \in \mathcal{T}_e$, then
    $u_{we}(\tau) = \min\{u_w(\tau) + 1, z(M_{b,w}, \tau)\}$
  endfor
  $u_{we}(T_e) = u_w(T_e) + 1$
  $M_{b,we} = M_{b,w}$
endif
7. $w = we$

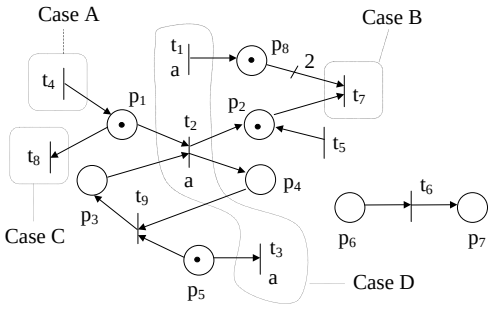

Figure 1: The generic substructure of a more complex Petri net that satisfies the contact-free assumption.

8. Goto 4. ■

Now, before examining in detail the steps of the algorithm, let us discuss the physical meaning of all the parameters characterizing the above set (1).

Let us preliminary observe that the firing of a nondeterministic transition t may be *detected* (or *reconstructed*) when a deterministic transition t_d is observed and the firing of t is strictly necessary to enable t_d. Therefore, using Algorithm 6, we define $M_{b,w}$ as the marking that we reach from the initial one by firing all the observed deterministic transitions, and all those nondeterministic transitions that have been detected. In the following we say that $M_{b,w}$ is the *basis marking* given the actual observation w.

Moreover, for each nondeterministic event e, the upper bound $u_w(T_e)$ denotes how many times the event e has been observed in w without being detected.

Finally, the upper bound $u_w(\tau)$ relative to a given subset $\tau \subset \mathcal{T}_e$, imposes a limit on the maximum number of times all transitions in τ may have fired, given the actual observation w, and taking into account that a certain number of nondeterministic transitions labeled e may have been detected.

Now, let us discuss in detail all cases in Algorithm 6. Consider the labeled Petri net in Figure 1 that represents the generic substructure of a more complex Petri net that satisfies the contact-free assumption (A4). Let us assume that in this subnet the only nondeterministic transitions are those labeled a. Let w be the actual observed word of events and let $M_{b,w}$ be the marking shown in Figure 1. Finally assume $|w|_a \geq 1$.

(Case A): *A deterministic transition t such that ${}^\bullet t \cap ({}^\bullet T^n {}^\bullet) = \emptyset$ fires.* If t_4 fires we only update the basis marking taking into account that the deterministic transition t_4 has fired, but we deduce no information on the number of times the nondeterministic transitions have eventually fired.

(Case B): *A deterministic transition t such that ${}^\bullet t \cap (T^n {}^\bullet) = P_t \neq \emptyset$ fires.* If t_7 fires, we know for sure that each place $p \in {}^\bullet t_7$ (namely, p_2 and p_8) contains a number of tokens that is greater or equal than $Pre(p, t_7)$. Now, given the basis marking $M_{b,w}$, if for some place $p \in {}^\bullet t_7$, $M_{b,w}(p) < Pre(p, t_7)$, we know for sure that the nondeterministic transition ${}^\bullet p$ has fired and we can also evaluate (see Algorithm 6) how many times it has fired. We consequently update the basis marking and the upper bounds relative to all subsets containing ${}^\bullet p$. In the case at hand, we can conclude that one of the previous observations of a was due to the firing of t_1. Therefore, the basis marking $M_{b,w}$ is updated to $M_{b,we} = M_{b,w} + C(\cdot, t_1) + C(\cdot, t_7)$.

(Case C): *A deterministic transition t such that ${}^\bullet t \cap ({}^\bullet T^n) \neq \emptyset$ fires.* If t_8 fires the upper bounds associated to subsets of nondeterministic transitions may decrease. In fact, if t_8 fires, we know for sure that if p is an input place of t_8, then it should contain a number of tokens that is greater or equal to $Pre(p, t_8)$. Therefore, if there is some nondeterministic transition exiting p, we know for sure that the maximum number of times it may have fired must ensure that in p there are at least $Pre(p, t_8)$ tokens. As an example, if in the actual case the upper bound associated to $\tau = \{t_2\}$ were 1, we reduce it to zero. Then, we update all the other $u_w(\tau)$'s relative to subsets τ containing t_2, as well as $u_w(T_a)$.

(Case D): *A nondeterministic event is observed.* If the nondeterministic event a is observed, we update the upper bounds $u_{wa}(\tau)$ relative to those subsets $\tau \in \mathcal{T}_a$ whose enabling degree at the current basis marking $M_{b,w}$ is greater than the bound $u_w(\tau)$. Furthermore, we always increment of one unity the value of the bound of T_a, i.e., $u_{wa}(T_a) = u_w(T_a) + 1$, that takes into account how many times the event a has been observed without being detected.

4 Control using $\mathcal{C}(w)$

The above linear algebraic characterization of the set of consistent markings can be efficiently used by a control agent to enforce a given specification on the plant behaviour. In particular, assume that the following hypothesis are verified.

(H1) The specification is given as a set of legal states of the form $\mathcal{L} = \{M \in \mathbb{N}^m \mid S^T \cdot M \leq \vec{k}\}$ where $S = [\vec{s}_1 \cdots \vec{s}_q]$ with $\vec{s}_j \in \mathbb{Z}^n$ and $\vec{k} = [k_1 \cdots k_q]$ with $k_j \in \mathbb{Z}$. This kind of specifications, called *generalized mutual exclusion constraints* (GMEC) are denoted $(S, \vec{k})$.

(H2) The controller may disable transitions to prevent the plant from entering a forbidden marking, computing a marking dependent control pattern $f(t, M) : T \times \mathbb{N}^m \to \{0, 1\}$. If $f(t, M) = 0$ then t is disabled by the controller at M, while if $f(t, M) = 1$ it is enabled.

(H3) All transitions are controllable, i.e., can be disabled by the controller.

It is well known that under the assumption that: the initial marking M_0 is legal, all transitions are controllable and the actual marking is known, an optimal (i.e., maximally permissive) control policy that enforces a given state specification is as follows.

Definition 7 (Optimal state feedback for GMEC)**.** *Given a GMEC $(S, \vec{k})$ and a marking M, the firing of transition t should be prevented from M if and only if leads from a legal to a forbidden marking, i.e.,*

$$f(t, M) = \begin{cases} 0 & \text{if } S^T \cdot M \leq \vec{k},\ M[t\rangle M',\ (\exists j)\ \vec{s}_j \cdot M' > k_j \\ 1 & \text{otherwise.} \end{cases}$$

■

When an observer is used in the control loop the actual marking M is not known and only a set of consistent markings $\mathcal{C} \subseteq \mathbb{N}^m$ is available to the controller. In this paper we discuss two different control laws, depending on the possibility of the controller to distinguish among transitions with the same label.

Label-based feedback (LBF). In this case the controller cannot distinguish among transitions that are undistinguishable by the observer (namely, transitions with the same label). Hence, if a transition labeled by event e is disabled, then the control pattern must simultaneously disable all other transitions labeled by e.

Transition-based feedback (TBF). In this case we assume that the controller is not constrained by the observer observation mask, and may assign a different control pattern to different transitions even if they share the same label e.

4.1 Label-based feedback

In this case the control law becomes a function $f(e, \mathcal{C}) : L \times 2^{\mathbb{N}^m} \to \{0, 1\}$ and can be given as follows.

Definition 8 (Optimal LBF for GMEC with observer)**.** *Given a GMEC $(S, \vec{k})$, a set of consistent markings $\mathcal{C} \subseteq \mathbb{N}^m$, and a λ-free labeling function $L : T \to E$, the firing of each transition labeled e should be prevented if and only if there exists a legal consistent marking M and a transition t labeled e, such that the firing of t from M leads to a forbidden marking, i.e.,*

$$f(e, \mathcal{C}) = \begin{cases} 0 & \text{if } (\exists M,\ \exists t : L(t) = e)\ M \in \mathcal{C},\ S^T \cdot M \leq \vec{k}, \\ & \quad M[t\rangle M',\ (\exists j)\ \vec{s}_j \cdot M' > k_j \\ 1 & \text{otherwise.} \end{cases}$$

■

The computation of the control law $f(e, \mathcal{C})$ may be carried out solving a number of linear integer programming problems (LIPP) of the form

$$\begin{cases} \max\ \vec{l}_j^T \cdot M' & \\ \text{s.t.} & \\ M \in \mathcal{C} & (a) \\ L^T \cdot M \leq \vec{k} & (b) \\ M \geq Pre(\cdot, t) & (c) \\ M' = M + C(\cdot, t) & (d) \\ M' \in \mathbb{N}^m & (e) \end{cases} \tag{3}$$

Equation (3) implies that all transitions labeled e are disabled if there exists at least one transition t labeled e that may fire – constraint (c) – and there exists a consistent marking M – constraint(a) – that is legal – constraint (b) – and from which the firing of t leads to a marking M' – constraint (d) – that is not legal because for at least one j it holds $h_j = \vec{s}_j^T \cdot M' > k_j$. Note that, as a consequence of Equations (1) and (2), constraint (a) is linear with respect to M.

Let us finally observe that in the case of label-based feedback, Algorithm 6 can still be used to compute the set of markings in which the *closed-loop system* may be, given the actual observation. In fact, if a nondeterministic event e is observed, this implies that a transition labeled e has fired (thus it was enabled by the controller). By assumption of LBF, this implies that all the other transitions labeled e were control enabled as well. Therefore, we may conclude that the controller does not influence the estimation procedure.

4.2 Transition-based feedback

When the controller may assign a different control pattern to nondeterminstic transitions with the same label the control pattern is a function $f(t, \mathcal{C}) : T \times 2^{\mathbb{N}^m} \to \{0, 1\}$ and can be given as follows.

Definition 9 (Optimal TBF for GMEC with observer)**.** *Given a GMEC $(S, \vec{k})$, a set of consistent markings $\mathcal{C} \subseteq \mathbb{N}^m$, and a λ-free labeling function $L : T \to E$, the firing of transition t should be prevented if and only if there exists a legal consistent marking M such that the firing of t from M leads to a forbidden marking, i.e.,*

$$f(t, \mathcal{C}) = \begin{cases} 0 & \text{if } (\exists M)\ M \in \mathcal{C},\ S^T \cdot M \leq \vec{k}, \\ & \quad M[t\rangle M',\ (\exists j)\ \vec{s}_j \cdot M' > k_j \\ 1 & \text{otherwise.} \end{cases}$$

■

The algorithm for the computation of the control law $f(t, \mathcal{C})$ is not given here for sake of brevity. It requires the solution of a certain number of LIPP analogous to that given by Equation (3).

It is important to observe that the possibility of assigning a different control pattern to nondeterministic transitions with the same label may reduce the nondeterminism of the net. In fact, if a nondeterministic event e is observed and only a subset of transitions $T'_e \subset T_e$ was enabled by the controller, we know for sure that none of the transitions in $T_e \setminus T'_e$ has fired. This implies that Algorithm 6 needs to be slightly modified when used to compute the set of markings in which the *closed-loop system* may be, given the actual observation. This point is formalized in the following algorithm.

Algorithm 10 (Upper b. and basis m. comp. under TBF)**.** When the controller implements a transition-based feedback, Case D of Algorithm 6 becomes:

let $T'_e = \{t \in T_e \mid f(t, \mathcal{C}) = 1\}$
if $T'_e = \{t\}$
  then $M_{b,we} = M_{b,w} + C(\cdot, t)$
  else
    for all $\tau \in \mathcal{T}_e$
      if $\tau \cap T'_e \neq \emptyset$
        then $u_{we}(\tau) = \min\{u_w(\tau) + 1, z(M_{b,w}, \tau)$

else $u_{we}(\tau) = u_w(\tau)$
endif
endfor
$u_{we}(T_e) = u_w(T_e) + 1$
$M_{b,we} = M_{b,w}$
endif ■

Therefore, in the case that there is only one transition t labeled e that is enabled by the controller, we update the basis marking and we do not increase the value of u_{we} (the event e is now deterministic). On the contrary, if there is more than one transition t labeled e that is enabled by the controller (i.e., $|T'_e| > 1$), we only update the upper bounds relative to the subsets of T_e containing transitions enabled by the controller, as well as the upper bound relative to T_e.

5 A final example

Let us consider a job-shop system consisting of 13 machines $\mathcal{M}_1$, $\mathcal{M}_2$, ..., $\mathcal{M}_{13}$, and 3 jobs J_1, J_2, J_3. Each job requires three different operations: the first operation provides semi-finished products (we assume an infinite availability of raw materials); the second operation provides finished products; finally, the third operation removes the finite product from the buffer. More precisely, the jobs must visit the machines in the following order:

$$J_1 = \{\mathcal{M}_1 \vee \mathcal{M}_4, \mathcal{M}_2, \mathcal{M}_3 \vee \mathcal{M}_8\},$$
$$J_2 = \{\mathcal{M}_5 \vee \mathcal{M}_{11}, \mathcal{M}_7, \mathcal{M}_9\},$$
$$J_3 = \{\mathcal{M}_{10} \vee \mathcal{M}_{13}, \mathcal{M}_{12}, \mathcal{M}_6\},$$

where $\vee$ denotes the logical *or*.

The operational state of each machine is detected by an appropriate symbol according to the following tables

$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6$	$\mathcal{M}_7$
a	b	c	d	e

$\mathcal{M}_8, \mathcal{M}_9, \mathcal{M}_{10}$	$\mathcal{M}_{11}$	$\mathcal{M}_{12}$	$\mathcal{M}_{13}$
f	g	h	i

Thus, we are not able to distinguish among the operations of machines $\mathcal{M}_4$, $\mathcal{M}_5$ and $\mathcal{M}_6$ (labeled by the same symbol d), and the operations of machines $\mathcal{M}_8$, $\mathcal{M}_9$ and $\mathcal{M}_{10}$ (labeled by the same symbol f).

The Petri net model of the job-shop system is sketched in Figure 2. The firing of transition t_i model the operational process of machine $\mathcal{M}_i$, while its label is the symbol associated to machine $\mathcal{M}_i$, for $i = 1, \ldots, 13$. Finally, the marking of places p_1, p_3 and p_5 represents the semi-finished product; the marking of places p_2, p_4 and p_6 represents the finished product.

Assume that the initial marking is that one reported in Figure 2, namely $M_0 = [0\ 0\ 0\ 0\ 0\ 1]^T$.

Let us first assume that no controller is added to the system.

Initially, when no event is observed the basis marking is the initial marking and all the upper bounds are set to zero. As a new event is observed, the algorithm updates the basis marking and the upper bounds as listed in Table 1. Note that for simplicity of notation in Table 1 we omitted the dependence of the bounds on w and denoted as u_i, $u_{i,j}$, $u_{i,j,k}$ the upper bounds relative to the subsets of transitions $\{t_i\}$, $\{t_i, t_j\}$ and $\{t_i, t_j, t_k\}$, respectively.

Data in the table are then used to construct the set of admissible markings as described in Theorem 5.

Let us show for instance how to use the table to compute the set $\mathcal{C}(d)$. It holds that $\mathcal{S}_d(d) = \{\vec{\sigma} = [\sigma_4\ \sigma_5\ \sigma_6]^T \in \mathbb{N}^3 \mid \sigma_4 \leq 1, \sigma_5 \leq 1, \sigma_6 \leq 1, \sigma_4 + \sigma_5 \leq 1, \sigma_4 + \sigma_6 \leq 1, \sigma_5 + \sigma_6 \leq 1, \sigma_4 + \sigma_5 + \sigma_6 = 1\}$. The solutions of this integer inequality system are: $\vec{\sigma}^1 = [0\ 0\ 1]^T$, $\vec{\sigma}^2 = [0\ 1\ 0]^T$, $\vec{\sigma}^3 = [1\ 0\ 0]^T$, which substituted in $M = M_{b,d} + C_d\vec{\sigma}^i$, $i = 1, 2, 3$ provide the set of admissible markings: $\mathcal{C}(d) =$

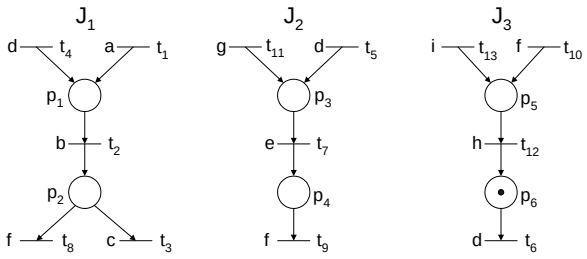

Figure 2: The labeled Petri net system considered in Section 5.

w	*Case*	$M_{b,w}{}^T$	u_4	u_5	u_6	u_{45}	$u_{4,6}$	$u_{5,6}$	$u_{4,5,6}$	u_8	u_9	u_{10}	$u_{8,9}$	$u_{8,10}$	$u_{9,10}$	$u_{8,9,10}$
ε		[0 0 0 0 0 1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	*D*	[0 0 0 0 0 1]	1	1	1	1	1	1	1	0	0	0	0	0	0	0
dd	*D*	[0 0 0 0 0 1]	2	2	1	2	2	2	2	0	0	0	0	0	0	0
ddi	*A*	[0 0 0 0 1 1]	2	2	1	2	2	2	2	0	0	0	0	0	0	0
ddie	*B*	[0 0 0 1 1 1]	2	1	1	1	2	1	1	0	0	0	0	0	0	0
ddieg	*A*	[0 0 1 1 1 1]	2	1	1	1	2	1	1	0	0	0	0	0	0	0
ddiegb	*B*	[0 1 1 1 1 1]	1	1	1	0	1	1	0	0	0	0	0	0	0	0
ddiegbd	*D*	[0 1 1 1 1 1]	2	2	1	1	2	2	1	0	0	0	0	0	0	0
ddiegbdf	*D*	[0 1 1 1 1 1]	2	2	1	1	2	2	1	1	1	1	1	1	1	1
ddiegbdfa	*A*	[1 1 1 1 1 1]	2	2	1	1	2	2	1	1	1	1	1	1	1	1
ddiegbdfab	*B*	[0 2 1 1 1 1]	2	2	1	1	2	2	1	1	1	1	1	1	1	1
ddiegbdfabc	*C*	[0 1 1 1 1 1]	2	2	1	1	2	2	1	1	1	1	1	1	1	1

Table 1: The evolution of the net in Figure 2 when no control law is acting on the system.

$\{[0\ 0\ 0\ 0\ 0\ 0]^T, [1\ 0\ 0\ 0\ 0\ 1]^T, [0\ 0\ 1\ 0\ 0\ 1]^T\}$. If we repeat the procedure for all the other events we obtain: $\mathcal{C}(dd) = \{[1\ 0\ 0\ 0\ 0\ 0]^T, [0\ 0\ 1\ 0\ 0\ 0]^T, [2\ 0\ 0\ 0\ 0\ 1]^T, [1\ 0\ 1\ 0\ 0\ 1]^T, [0\ 0\ 2\ 0\ 0\ 1]^T\}$, $\mathcal{C}(ddi) = \{[1\ 0\ 0\ 0\ 1\ 0]^T, [0\ 0\ 1\ 0\ 1\ 0]^T, [2\ 0\ 0\ 0\ 1\ 1]^T, [1\ 0\ 1\ 0\ 1\ 1]^T, [0\ 0\ 2\ 0\ 1\ 1]^T\}$, and so on.

Now, assume that a controller is used to enforce a given specification. Let assumptions (H1) to (H3) be verified.

Let us first assume that all finite products are stored in a finite capacity buffer, and let 3 be its capacity. The set of legal markings is

$$\mathcal{L} = \{M \in \mathbb{N}^m \mid M(p_2) + M(p_4) + M(p_6) \leq 3\}.$$

The closed-loop behaviour of the system is obviously the same in the case of LBF and in the case of TBF because the set of transitions that may potentially violate the constraint is $J_t = \{t_2, t_7, t_{12}\} \subset T^d$, i.e., it only contains deterministic transitions. Thus the nondeterministic transitions are always enabled by the controller, because regardless of the actual observation, their firing may never lead to a violation of the constraint.

It is easy to verify that the previously considered word $w = ddiegbdfabc$ cannot be observed. More precisely, for all prefixes $w' \preceq ddiegb$ all transitions are enabled by the controller. On the contrary, after a further event d occurs all transitions in J_t are disabled by the controller, and the control patterns keep unaltered until the word $w = ddiegbdfa$ is observed, thus transition t_2 cannot fire (the event b cannot be observed).

Now, assume that all finite products are stored in an infinite capacity buffer, while semi-finished products relative to jobs J_1 and J_3 are stored in a finite capacity buffer, and let 2 be its capacity. The set of legal markings is

$$\mathcal{L} = \{M \in \mathbb{N}^m \mid M(p_1) + M(p_5) \leq 2\}.$$

In such a case the closed-loop behaviour of the system is no longer the same in the case of LBF and in the case of TBF because the set of transitions that may potentially violate the constraint is $J_t =$

$\{t_1, t_4, t_{10}, t_{13}\}$, i.e., it also contains nondeterministic transitions.

A very simple example of this fact is given by the word $w = ddd$. In the case of TBF this sequence of events may occur, while in the case of LBF it cannot occur. In fact, when the first event d is observed all transitions are enabled (both in the case of TBF and LBF), thus a second event d may occur. At this point, the set $\mathcal{C}(dd)$ contains the marking $M = [2\ 0\ 0\ 0\ 0\ 1]^T$ that enables a transition labeled d, namely t_4, whose firing would lead to a violation of the constraint. In the case of the LBF this implies that all transitions labeled d are disabled by the controller, thus a further event d cannot occur. On the contrary, in the case of the TBF, transition t_4 is disabled by the controller, but both transitions t_5 and t_6 may fire, thus a further event d may occur.

6 Conclusions and future work

The contribution of this paper is twofold.

We first presented a marking estimation procedure that can be applied to λ-free labeled Petri nets (the formal proof of the correctness of this procedure can be found in [6]). In particular, under the assumption that all nondeterministic transitions are contact-free, the set of markings consistent with an observed word can be described by a constraint set of linear inequalities that has a fixed structure that does not change as the length of the observed sequence increases.

Then, we shown how this marking estimation can be used by a control agent to enforce some specifications. In particular we discussed two different cases, depending on the possibility of the controller to assign a different control pattern to transitions that share the same label.

We plan to extend our results in several ways.

Firstly, we plan to remove the contact-free assumption, allowing the subnet composed of the nondeterministic transitions to have a more general structure.

Secondly, we believe it may be possible to modify the structure of the constraint set to also take into account the case that the initial marking is not known.

Finally, we plan to extend this approach to *arbitrary labeling functions*, i.e., functions $L : T \to E \cup \{\varepsilon\}$ that may assign to one or more transitions the empty string ε.

References

[1] P.E. Caines, R. Greiner, S. Wang, (1988). "Dynamical logic observers for finite automata," *27th Conf. on Decision and Control*, Austin, Texas, pp. 226–233.

[2] P.E. Caines, S. Wang, (1989). "Classical and logic based regulator design and its complexity for partially observed automata," *28th Int. Conf. on Decision and Control*, Tampa, Florida, pp. 132–137.

[3] D. Corona, A. Giua, J. Júlvez, C. Seatzu, (2003). "Observers for nondeterministic λ-free labeled Petri nets," *9th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Lisbon, Portugal, pp. 307–314.

[4] S. Gaubert, A. Giua, (1999). "Petri net languages and infinite subsets of $\mathbb{N}^m$," *Journal of Computer and System Sciences*, Vol. 59, No. 3, pp. 373-391.

[5] A. Giua, C. Seatzu, (2002). "Observability of place/transition nets," *IEEE Trans. on Automatic Control*, Vol. 47, No. 9, pp. 1424-1437.

[6] A. Giua, D. Corona, C. Seatzu, (2004). "State Estimation of λ-free Labeled Petri Nets with Contact-Free Nondeterministic Transitions," *Technical Report*, University of Cagliari, Italy. Also submitted to *Discrete Event Dynamic Systems.*

[7] R. Kumar, V. Garg, S.I. Markus, (1993). "Predicates and predicate transformers for supervisory control of discrete event dynamical systems," *IEEE Trans. on Automatic Control*, Vol. 38, No. 2, pp. 232-247.

[8] M.E. Meda, A. Ramírez, A. Malo (1998). "Identification in discrete event systems," *IEEE Int. Conf. on Systems, Man and Cybernetics*, San Diego, California, pp. 740-745.

[9] T. Murata, (1989). "Petri nets: properties, analysis and applications," *Proc. IEEE*, Vol. 77, No. 4, pp. 541-580.

[10] C.M. Özveren, A.S. Willsky, (1990). "Observability of discrete event dynamic systems," *IEEE Trans. on Automatic Control*, Vol. 35, No. 7, pp. 797-806.

[11] J.L. Peterson, (1981). *Petri net theory and the modeling of systems*, Prentice-Hall.

[12] P.J. Ramadge, (1986). "Observability of discrete-event systems," *25th Int. Conf. on Decision and Control*, Athens, Greece, pp. 1108-1112.

[13] L. Zhang, L.E. Holloway, (1995). "Forbidden state avoidance in controlled Petri nets under partial observation," *33rd Allerton Conference*, Monticello, Illinois, pp. 146–155.