Observer based state-feedback control of timed Petri nets with deadlock recovery

Alessandro Giua, Carla Seatzu

Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari, Italy

Phone: +39-070-675-5751 – Fax: +39-070-675-5782 – Email: {giua,seatzu}@diee.unica.it

Francesco Basile

Dip. di Ing. dell'Informazione e Ing. Elettrica, Università di Salerno, Italy

Phone: +39-089-96-4400 – Fax: +39-06-233 227 957 – Email: fbasile@unisa.it

Abstract

This paper discusses the problem of controlling a timed Petri net whose marking cannot be measured but is estimated using an observer. The control objective is that of enforcing a set of generalized mutual exclusion constraints (GMEC) and all transitions are assumed to be controllable. We show that the use of marking estimates may significantly reduce the performance of the closed-loop system and in particular may lead to a deadlock. Firstly, we present a linear algebraic characterization of deadlock markings based on siphon analysis. Secondly, we show how this characterization may be used to derive a procedure that may be invoked to recover from a controller induced deadlock. Finally, we assume that the timing delays associated to transitions are known and show how this knowledge can be used to improve the marking estimate and to recover the net from partial deadlocks. This procedure is similar to the one used for deadlock recovery and may be invoked whenever a transition has not fired for a time longer than its expected delay.

Published as: A. Giua, C. Seatzu, F. Basile, "Observer-based state feedback control of timed Petri nets with deadlock recovery," *IEEE Trans. on Automatic Control*, Vol. 49, No. 1, pp. 17-29, January 2004.

1 Introduction

In this paper we deal with the issue of controlling a Petri net whose marking cannot be measured. The problem of controlling a discrete event system under incomplete information has often been discussed in the literature. As an example, the use of state-feedback control under partial *state* observation has been discussed by Li and Wonham [9, 10] and by Takai *et al.* [19]. In the work of these authors the partial observation is due to a static mask, that maps the plant state space into an observation space. In the Petri nets framework we also mention the work of Zhang and Holloway [21] that used a Controlled Petri Net model for forbidden state avoidance under partial *event* observation with the assumption that the initial marking be known. Moody and Antsaklis have also discussed the controller design of monitor places for nets with uncontrollable and unobservable transitions [12].

The approach we develop in this paper is based on the classical system theory notion of a state-feedback controller that uses an observer to estimate the plant state. In previous works [7] we have shown how it is possible to estimate the actual marking of the net based on the observation of a word of events (i.e., transition firings) and an algorithm was given for computing the marking estimate. The estimate is always a lower bound of the actual marking. The system that computes the estimate is called an observer. The special structure of Petri nets allows us to use a simple linear algebraic formalism for estimate computation. In particular, the set $\mathcal{C}(\sqsupseteq)$ of *markings consistent with an observed word* w, i.e., the set of markings in which the system may actually be given the observed word, can easily be described in terms of the observer estimate and can be characterized as the integer solutions of a linear constraint set. Other approaches to the design of Petri net observers can also be found in [16].

In [7] we have also shown how the estimate generated by the observer may be used to design a state feedback controller, that ensures that the controlled system never enters a set of forbidden states. We considered a special class of safety specifications that limit the weighted sum of markings in subsets of places called generalized mutual exclusion constraints (GMEC).

Clearly, the use of marking estimates, as opposed to the exact knowledge of the actual marking of the plant, leads to a worse performance of the closed-loop system. In fact, in a safety problem the aim of the controller is that of preventing all those transition firings that lead to a forbidden marking. If the actual marking is not exactly known, but is only known to belong to a given consistent set $\mathcal{C}$, the controller must forbid all transitions firing that from "any" marking in $\mathcal{C}$ may lead to a forbidden marking and the controller

becomes usually more restrictive as the cardinality of this set increases. Because of this it may be the case that the controlled system reaches a deadlock, i.e., a blocking condition, even if it is deadlock free when perfect information about the marking is available.

We first show that, using siphon analysis, the set of deadlock markings $\mathcal{M}_b$ of a structurally bounded net can be characterized as the integer solution of a linear constraint set. Siphon analysis has been already used by several authors to derive deadlock avoidance policies: see [1, 4, 5, 14]. The approach we present here is different from the above mentioned approaches in two ways. Firstly, our approach only aims to give a characterization of deadlock markings. On the contrary, the referenced approaches aim to solve a more complex problem, namely that of deriving a deadlock avoidance policy: to do this it necessary to also characterize *impending deadlock markings*, i.e., markings that are not dead but that will lead to a deadlock in a finite number of steps. Secondly, since we solve a less complex problem we are able to derive a simpler (in terms of number of constraints and number of unknowns) characterization that applies to a large class of nets (ordinary and structurally bounded), while the referenced approaches are only valid for restricted classes of nets.

Then, we focus our attention to timed Petri nets, i.e., Petri nets where a delay is associated to each transition. The delay represents the time that must elapse from the enabling of the transition until it fires.

We initially assume that a very loose information on the timing structure is available. More precisely, we assume that if no transition firing occurs within a reasonable amount of time in a controlled system — we say that the *net has timed out* — one can conclude that a deadlock has occurred and a recovery procedure should be invoked. The characterization based on siphon analysis may be used to derive a recovery procedure from deadlocks induced by the observer.

We also explore the characterization of those cases in which the proposed procedure works. More precisely, we consider a particular class of macromarkings and derive a sufficient condition to ensure that the controlled net will never time out. We also give a sufficient condition to ensure that, in the case that a time-out occurs, the proposed procedure will always recover the net from a deadlock. Finally, we show how the linear algebraic characterization of deadlock markings may also be used to improve the marking estimate, thus providing a better characterization of the set of consistent markings.

In a final part of the paper, we consider the case in which the timing structure is known and propose a new control algorithm that uses the previous marking estimate and control approach, but that also takes into account the knowledge of the delays and of the enabling status of each transition. This algorithm should be invoked whenever a

transition has not fired for a time larger than its expected delay, i.e., when a *transition has timed out*. Thus it not only allows the supervisor to recover from total deadlocks (as in the previous case) but it allows one to detect partial deadlocks as well, and in general it improves and accelerates the convergence of the marking estimation procedure. We also show how the observer can use this information to restrict the set of consistent markings.

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 [13].

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; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ 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 $M : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. In the following 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.

A transition t is *marking enabled* at M if $M \geq Pre(\cdot, t)$. In this paper we also assume that a *supervisor*, i.e., an external control agent, may forbid the occurrence of a transition specifying a marking dependent control pattern $f(t, M) : T \times \mathbb{N}^m \rightarrow \{0, 1\}$ such that

$$f(t, M) = \begin{cases} 1 & \text{if } t \text{ is } \textit{control enabled}, \\ 0 & \text{if } t \text{ is } \textit{control disabled}. \end{cases}$$

A transition t is *enabled* at M if it is marking enabled and control enabled. A transition t enabled at M may *fire*, yielding the marking $M' = M + C(\cdot\,, t)$.

We write $M\ [w\rangle\ M'$ to denote that the enabled sequence of transitions w may fire at M yielding M', or equivalently we use the notation $M' = w(M)$ and $M = w^{-1}(M')$. Moreover, we denote $w(M_0) = M_w$. Finally, we denote ε the sequence of null length. The set of all sequences firable in $\langle N, M_0 \rangle$ is denoted $L(N, M_0)$ (this is also called the prefix-closed free language of the net). If the firing sequence w is enabled at M_0, we also say that w is a word in $L(N, M_0)$.

A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence w such that $M_0\ [w\rangle\ M$. 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 nonnegative integer vector $\vec{x} \neq \vec{0}_m$ such that $\vec{x}^{\,T} \cdot C = \vec{0}_n^{\,T}$ is called a *P–invariant* (here $\vec{0}_k$ denotes a $k \times 1$ vector of zeros). A P-invariant is *minimal* if there does not exist a P-invariant $\vec{y}$ such that $\vec{y} \leq \vec{x}$.

A transition t is said to be *live* if for any $M \in R(N, M_0)$, there exists a sequence of transitions firable from M which contains t. A Petri net is said to be live if all transitions are *live*.

A marking M is a *deadlock* (or *dead*) marking if no transition $t \in T$ may fire at M. A Petri net is said to be *deadlock–free* if at least one transition is enabled at every reachable marking.

A place p is said to be *bounded* if there exists a constant k such that $M(p) \leq k$ for all $M \in R(N, M_0)$. A net system is bounded if all places are bounded. A net is *structurally bounded* if it is bounded for all initial markings.

A P/T net is called *ordinary* when all of its arc weights are 1's. A *siphon* of an ordinary net is a *non–empty* set of places $\mathcal{S} \subseteq P$ such that: $\bigcup_{p \in \mathcal{S}} {}^\bullet p \subseteq \bigcup_{p \in \mathcal{S}} p^\bullet$. A siphon is *minimal* if it is not the superset of any other siphon. In the following we denote as $\vec{s} \in \{0,1\}^m$ the characteristic vector of $\mathcal{S}$, where $s_i = 1$ if place $p_i \in \mathcal{S}$ and $s_i = 0$ otherwise.

Definition 1. 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 restriction of $Pre, Post$ to T'. The net N' can also be thought as obtained from N by removing all transitions in $T \setminus T'$. We also write $N' \prec_{T'} N$. ■

A deterministic *timed* P/T net is a pair (N, δ), where $N = (P, T, Pre, Post)$ is a standard P/T net, and $\delta(t) : T \to \mathbb{R}_0^+$, called release delay, assigns a non-negative fixed firing duration to each transition. A transition with a release delay equal to 0 is said to be immediate. The value of $\delta(t)$ represents the time that must elapse, starting from the time at which the transition t is enabled, until it fires. We use single server-semantics, i.e., no concurrent firings of the same transition are possible.

3 Marking estimation with macromarkings

In previous works [7] the authors dealt with the problem of reconstructing the marking of a P/T net assuming that partial information about the initial marking is available in the form of a *macromarking*.

Definition 2 ([7])**.** Assume that the set of places P can be written as the union of $r+1$ subsets: $P = P_0 \cup P_1 \cup \cdots \cup P_r$ such that $P_0 \cap P_j = \emptyset$, for all $j > 0$. The number of tokens contained in P_j $(j > 0)$ is known to be b_j, while the number of tokens in P_0 is unknown. For each P_j, let $\vec{v}_j$ be its characteristic vector, i.e., $v_j(p) = 1$ if $p \in P_j$, else $v_j(p) = 0$.

The *macromarking defined by* $V = [\vec{v}_1, \cdots, \vec{v}_r]$ *and* $\vec{b} = [b_1, \cdots, b_r]$ is the set of markings $\mathcal{V}(V, \vec{b}) = \{M \in \mathbb{N}^m \mid V^T M = \vec{b}\}$. ■

The notion of macromarking occurs frequently when describing systems containing a known set of resources (e.g., parts, machines) whose actual conditions (e.g., exact location of parts within the plant, state of a machine) is unknown.

We make the following assumptions.

A1) The structure of the net $N = (P, T, Pre, Post)$ is known, while the initial marking M_0 is not.

A2) The event occurrences (i.e., the transition firings) can be observed.

A3) The initial marking M_0 belongs to the macromarking $\mathcal{V}(V, \vec{b})$, i.e., it satisfies the equation $V^T M_0 = \vec{b}$.

We also introduce the following notation.

Definition 3 ([7])**.** After the word w has been observed we define the set of $w-$consistent markings as the set of all markings in which the system may be given the observed behavior and the initial marking, i.e., the set $\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \exists M_0 \in \mathcal{V}(V, \vec{b}),\ M_0[w\rangle M\}$. ■

Given an evolution of the net $M_0[t_{\alpha_1}\rangle M_1[t_{\alpha_2}\rangle \cdots$, we use the following algorithm to compute the estimate μ_w of each actual marking M_w based on the observation of the word of events $w = t_{\alpha_1} t_{\alpha_2} \cdots t_{\alpha_k}$, and of the knowledge of the initial macromarking $\mathcal{V}(V, \vec{b})$. The same algorithm also enables us to compute the bound B_w, depending on the word w and on the initial macromarking $\mathcal{V}(V, \vec{b})$, used to characterize the set of consistent markings $\mathcal{C}(w)$.

Algorithm 4 ([7])**. (Marking Estimation with Event Observation and Initial Macromarking).**

1. Let the current observed word be $w = \varepsilon$ (the empty string).
2. Let the initial estimate be μ_ε, with $\mu_\varepsilon(p) = \min\{M(p) \mid V^T \cdot M = \vec{b}\}$.
3. Let the initial bound be $B_\varepsilon = \vec{b} - V^T \cdot \mu_\varepsilon$.
4. Wait until a new transition, say t, fires.
5. Update the estimate μ_w to μ'_{wt} with $\mu'_{wt}(p) = \max\{\mu_w(p), Pre(p, t)\}$.
6. Let $\mu_{wt} = \mu'_{wt} + C(\cdot, t)$.
7. Let $B_{wt} = B_w - V^T \cdot (\mu'_{wt} - \mu_w)$.

8. Goto 4. ■

In simple words, if the currently observed word is w and transition t fires, the algorithm firstly updates the current estimate from μ_w to μ'_{wt} adding the minimal number of tokens required to enable t. Secondly, the algorithm computes μ_{wt} as the marking obtained from μ'_{wt} firing t.

The set of consistent markings can be characterized in terms of the estimate μ and bound B[1] as follows.

Theorem 5 ([7])**.** Given a net with initial macromarking $\mathcal{V}(V, \vec{b})$, an observed word $w \in L(N, M_0)$, and the corresponding estimated marking μ_w and bound B_w computed by Algorithm 4, the set of $w-$*consistent markings* coincides with the set of $(\mu_w, B_w)-$*consistent markings*, i.e.,

$$\mathcal{C}(w) = \mathcal{M}(\mu_w, B_w) \stackrel{\text{def}}{=} \{M \in \mathbb{N}^n \mid M \geq \mu_w,\ V^T \cdot M = V^T \cdot \mu_w + B_w\}. \tag{1}$$

■

4 Control using observers

In this section we show how the marking estimate constructed with the formalism discussed in the previous section can be used by a control agent to enforce a given specification on the plant behavior.

We make several assumptions that are briefly discussed here.

- We assume that the specification on the desired behavior is given as a set of legal markings $\mathcal{L}$. The use of marking (i.e., state) specifications leads naturally to the design of a state feedback control law [8] that may be easily adapted to the presence of an observer in the feedback loop.

- We consider a special type of state specifications called *generalized mutual exclusion constraints* (GMEC) that have been considered by various authors [6, 11, 20].

  Given an integer matrix $L = [\vec{l_1} \cdots \vec{l_q}]$ with $\vec{l_j} \in \mathbb{Z}^m$ and a vector $\vec{k} = [k_1, \cdots, k_q]$ with $k_j \in \mathbb{Z}$, a GMEC $(L, \vec{k})$ defines the set of legal markings $\mathcal{L} = \{M \in \mathbb{N}^m \mid L^T \cdot M \leq \vec{k}\}$.

- The controller may disable transitions to prevent the plant from entering a forbidden marking, computing a marking dependent control pattern $f(t, M) : T \times \mathbb{N}^m \rightarrow$

[1] To avoid a heavy notation, we will drop the subscript w from μ and B whenever it is possible without introducing ambiguity.

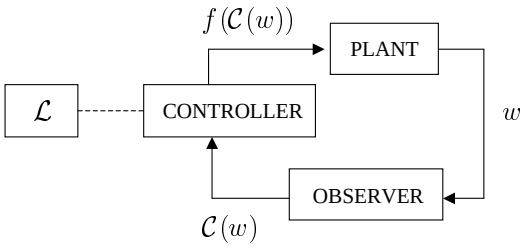

Figure 1: *State feedback control loop with observer.*

$\{0,1\}$. If $f(t,M)=0$ then t is disabled by the controller at M, while if $f(t,M)=1$ it is enabled.

- All transitions are controllable, i.e., can be disabled by the controller.

The considered control scheme is shown in Figure 1.

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 6 (Optimal state feedback for GMEC)**.** Given a GMEC $(L,\vec{k})$ 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.,

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

■

When an observer is used in the control loop, the actual marking M is not known and only the set of consistent markings $\mathcal{C}\subseteq\mathbb{N}^m$ is available to the controller. The control law now becomes a function $f(t,\mathcal{C}):T\times 2^{\mathbb{N}^m}\to\{0,1\}$ and can be given as follows.

Definition 7 (Optimal state feedback for GMEC with observer)**.** Given a GMEC $(L,\vec{k})$ and a set of consistent markings $\mathcal{C}\subseteq\mathbb{N}^m$, 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.,

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

■

The computation of the control pattern may be carried out solving a number of linear integer programming problems (IPP) as given in the following algorithm.

Algorithm 8 (Computation of the optimal state feedback with observer)**.** The control law in Definition 7 can be computed as follows.

1. `For all` $t \in T$, `let` $J_t = \{j \mid \vec{l_j}^T \cdot C(\cdot, t) > 0\}$ `be the set of indices of those constraints that may potentially be violated by the firing of` t.

2. `Solve for each` $j \in J_t$ `the IPP`

$$\begin{cases} \max \ \vec{l_j}^T \cdot M' & \\ \texttt{s.t.} & \\ M \in \mathcal{C} & (a) \\ L^T \cdot M \leq \vec{k} & (b) \\ M \geq Pre(\cdot, t) & (c) \\ M' = M + C(\cdot, t) & (d) \\ M' \in \mathbb{N}^m & (e) \end{cases} \tag{2}$$

   `and let` $h_j(t)$ `be its optimal solution.`

3. `Define`

$$f(t, \mathcal{C}) = \begin{cases} 0 & \texttt{if } (\exists j \in J_t) h_j(t) > k_j \\ 1 & \texttt{otherwise.} \end{cases} \tag{3}$$

■

Thus a transition t is disabled only if it 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 $h_j(t) = \vec{l_j}^T \cdot M' > k_j$. Note that, as a consequence of Equation (1), constraint (a) is linear with respect to M.

Finally, we state an elementary proposition that will be used in the following.

Proposition 9. Let $\mathcal{C}'$ and $\mathcal{C}''$ be two sets of consistent markings, with $\mathcal{C}' \subseteq \mathcal{C}''$. Then $f' = f(\cdot, \mathcal{C}')$ *is at least as permissive as* $f'' = f(\cdot, \mathcal{C}'')$ i.e., for all t it holds $f(t, \mathcal{C}') \geq f(t, \mathcal{C}'')$. We denote this writing $f' \geq f''$.

Proof. For all t and for all j, $\mathcal{C}' \subseteq \mathcal{C}''$ implies $h'_j(t) \leq h''_j(t)$, where $h'_j(t)$ and $h''_j(t)$ denote the solutions of (2) with, resp., $\mathcal{C} = \mathcal{C}'$ and $\mathcal{C} = \mathcal{C}''$. Thus the result follows from the definition of f given in (3). □

A trivial consequence of this proposition is the following. If the actual marking M is perfectly known the set of consistent markings is $\mathcal{C}' = \{M\}$. If the actual marking can only be estimated by an observer, then the set of consistent markings is $\mathcal{C}'' \supseteq \mathcal{C}'$. This

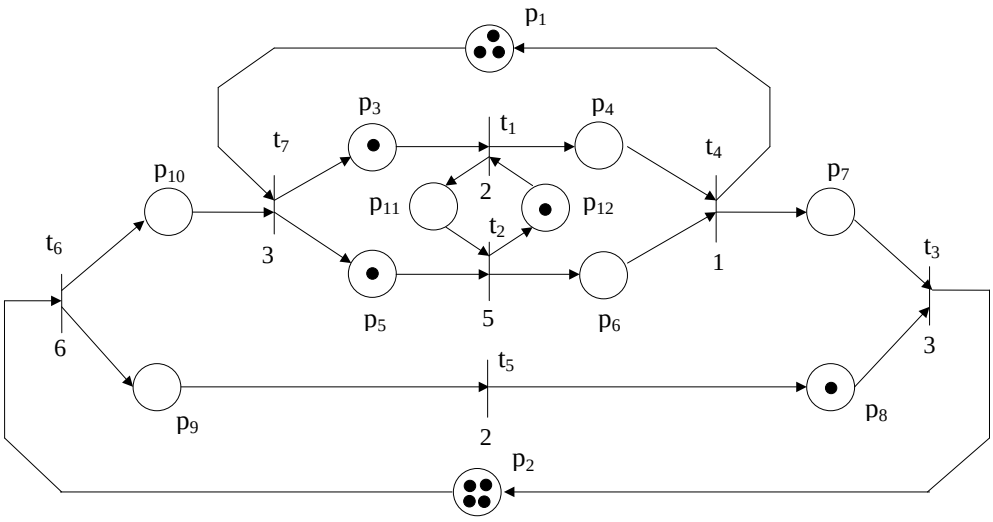

Figure 2: *Petri net model of the assembly system.*

means that the control pattern computed using an observer may be more restrictive than the optimal state feedback computed when the actual marking is known. As shown in the following example this may often lead to a block.

4.1 A manufacturing example

Now, let us apply the above methodology to a manufacturing system whose Petri net model is shown in Figure 2.

This assembly system, that is similar to the one described in [15], consists of five machines, $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$, $\mathcal{M}_4$ and $\mathcal{M}_5$ whose operational process is modeled by the firing of transitions t_1, t_2, t_3, t_4 and t_5, respectively. Two principal types of operations are involved in this manufacturing system: *regular operations* and *assembly operations*. Regular operations (modeled by transitions t_1, t_2 and t_5) just transform a component of the intermediate product. Assembly operations (modeled by transitions t_3 and t_4) put components together to obtain a more complex component of a final product or the final product itself. Note that this model uses transitions (t_6 and t_7) which do not represent operations but the beginning of the manufacturing of components which are required to assemble a more complex component or the final product. In this example there are two manufacturing levels, the primary one, performed by $\mathcal{M}_3$, leads to finite product, the secondary one, performed by $\mathcal{M}_4$, leads to semi–finished (in–working) product.

The markings of places p_1 and p_2 represent the number of assembly servers for t_4 and t_3 respectively. The marking of places p_3, p_5, and p_9 represent the availability of parts to

be processed (raw materials), while the marking of places p_4, p_6, p_7 and p_8 represent the availability of semi–finished products. Places p_{11} and p_{12} ensure that machines $\mathcal{M}_1$ and $\mathcal{M}_2$ work alternatively.

The Petri net model in Figure 2 is a strongly connected event graph with $m = |P| = 12$ and $n = |T| = 7$. There exist ten elementary circuits, that correspond to an equal number of P-invariants. If we assume that the initial marking of the net is $M_0 = [3\ 4\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$, we have (here to avoid a heavy notation we denote as M_i the marking of place p_i)

$$\begin{cases} M_2 + M_4 + M_5 + M_7 + M_{10} + M_{12} = 6 & (1) \\ M_2 + M_3 + M_6 + M_7 + M_{10} + M_{11} = 5 & (2) \\ M_2 + M_5 + M_6 + M_7 + M_{10} = 5 & (3) \\ M_2 + M_3 + M_4 + M_7 + M_{10} = 5 & (4) \\ M_2 + M_8 + M_9 = 5 & (5) \\ M_1 + M_4 + M_5 + M_{12} = 5 & (6) \\ M_1 + M_3 + M_6 + M_{11} = 4 & (7) \\ M_1 + M_5 + M_6 = 4 & (8) \\ M_1 + M_3 + M_4 = 4 & (9) \\ M_{11} + M_{12} = 1 & (10) \end{cases} \tag{4}$$

We assume that the above set of P-invariants coincides with the macromarking, thus $B_\varepsilon = \vec{b} = [6\ 5\ 5\ 5\ 5\ 5\ 4\ 4\ 4\ 1]^T$.

Moreover, we assume that the controller must enforce two specifications:

$$\begin{cases} M_3 + M_5 \leq 3 & (a) \\ M_9 \leq 3. & (b) \end{cases} \tag{5}$$

The first specification requires that at most 3 raw parts may be simultaneously waiting to be processed by either machine $\mathcal{M}_1$ or $\mathcal{M}_2$. The second specification requires that at most 3 raw parts may be waiting to be processed by machine $\mathcal{M}_5$.

Finally, we assume that the delay times associated to transitions are those shown in Figure 2.

If the marking of the net is measurable, then the controlled net is live, as it can be verified by reachability analysis. On the contrary, if the marking of the plant is not measurable, an observer must be used in the control loop and this leads to an observer-induced deadlock. The closed loop behavior is that shown in Figure 3 where each node is labeled with $(M/\mu/B)$ and where for each marking the set $T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}$ of transitions disabled by the controller is shown. Finally, variable *now* denotes the actual value of time.

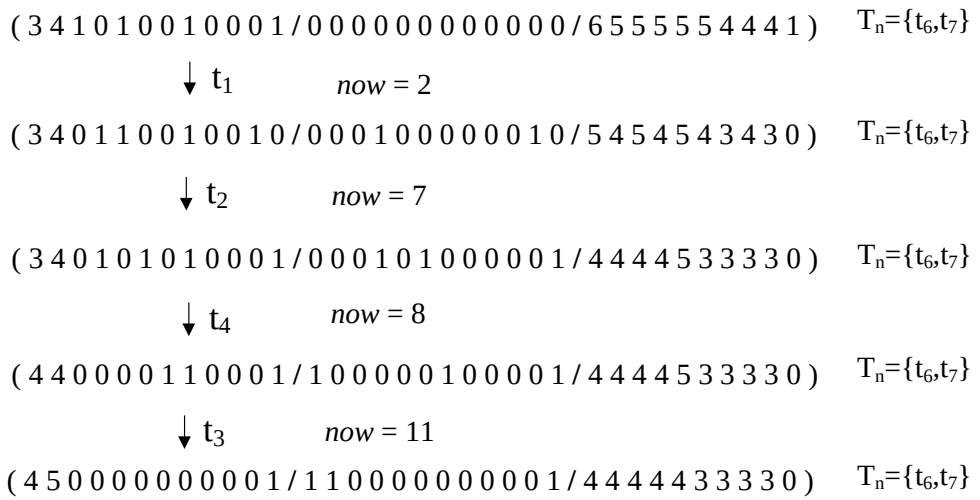

Figure 3: *The evolution of the net in Figure 2 when no deadlock recovery procedure is applied.*

After the sequence $w = t_1t_2t_4t_3$ has fired, only transition t_6 is marking enabled. Nevertheless, the controller prevents its firing because there exists at least one marking in $\mathcal{C}(t_1t_2t_4t_3)$ from which the firing of t_3 would potentially violate specification (b). Note that the controller also prevents the firing of t_7 because its firing would potentially violate specification (a).

5 A linear characterization of deadlock markings

In this section we present a linear algebraic characterization of deadlock markings based on siphon analysis that will be used in the next section to derive a procedure for deadlock recovery. Such a characterization is valid for ordinary and structurally bounded Petri nets. Siphon based techniques for deadlock analysis and avoidance have also been used by other authors [1, 4, 5, 14].

We firstly observe that, by definition, the characteristic vector $\vec{s}$ of a siphon $\mathcal{S}$ is such that:

$$\forall\, t \in T \quad Post^T(\,\cdot\,, t) \cdot \vec{s} > 0 \;\Rightarrow\; Pre^T(\,\cdot\,, t) \cdot \vec{s} > 0. \tag{6}$$

Condition (6) means that if p_i belongs to a siphon $\mathcal{S}$ (i.e., $s_i = 1$) and t inputs in p_i (i.e., $Post(p_i, t) > 0$), then there must exist at least one place p_j in the siphon (i.e., $s_j = 1$) inputting in t (i.e., $Pre(p_j, t) > 0$).

Condition (6) can also be rewritten as a nonlinear inequality:

$$\forall\, t \in T \quad sign(Pre^T(\,\cdot\,, t) \cdot \vec{s}) \geq sign(Post^T(\,\cdot\,, t) \cdot \vec{s}) \tag{7}$$

where $sign(\vec{x})$ is a vector whose i–th component is 1 (resp., 0, -1) if the i–th component of $\vec{x}$ is positive (resp., null, negative).

Since the above inequality holds for all t, we can write

$$sign(Pre^T \cdot \vec{s}) \geq sign(Post^T \cdot \vec{s}). \tag{8}$$

We can finally state the following result.

Lemma 10. A set of places $\mathcal{S} \subseteq P$ is a siphon of the net $N = (P, T, Pre, Post)$ if and only if its characteristic vector $\vec{s}$ is such that

$$K_1 \cdot Pre^T \cdot \vec{s} \geq Post^T \cdot \vec{s}, \tag{9}$$

where $K_1 = \max_{t \in T} Post^T(\,\cdot\,, t) \cdot \vec{1}$.

Proof. We observe that $Post^T(\,\cdot\,, t) \cdot \vec{s} \leq Post^T(\,\cdot\,, t) \cdot \vec{1} \leq K_1$. Thus a vector $\vec{s} \in \{0,1\}^m$ is a solution of (9) if and only if it is a solution (8). □

Secondly, let $M \in \mathbb{N}^m$ be a generic marking of N. If M corresponds to a reachable marking of the net such that the siphon $\mathcal{S}$ with characteristic vector $\vec{s}$ is empty, then

$$M^T \cdot \vec{s} = 0. \tag{10}$$

For structurally bounded Petri nets, equation (10) can be easily converted into a linear equivalent equation.

Lemma 11. Given a structurally bounded net $N = (P, T, Pre, Post)$, a siphon $\mathcal{S}$ with characteristic vector $\vec{s}$ is empty at marking M if and only if

$$K_2 \cdot \vec{s} + M \leq K_2 \cdot \vec{1}_m, \tag{11}$$

where K_2 is a positive integer. More precisely, K_2 should be chosen greater or equal to the maximum structural bound of p, for $p \in P$ [18], where structural bounds can be determined by using any LP software.

Proof. Equation (11) implies that if for a given j, $s_j = 0$ (i.e., place p_j does not belong to the siphon) then no constraint exists on the marking of p_j, since the equation $M(p_j) \leq K_2$ is satisfied for all reachable markings. On the contrary, if $s_j = 1$ (i.e., place p_j belongs to the siphon) then p_j must be empty. □

An analogous linear characterization of siphons has been already proposed by Chu and Xie in [4].

Thirdly and finally, to completely characterize the set of deadlock markings we use the following results that apply to *autonomous*, i.e., uncontrolled, nets.

Lemma 12 ([17])**.** Let N be an ordinary marked net. If $M \in \mathbb{N}^m$ is a dead marking then the set of empty places $\mathcal{S}_\emptyset = \{p \in P \mid M(p) = 0\}$ is an unmarked siphon of N.

We restate the previous result in a slightly different form.

Lemma 13. Let N be an ordinary marked net. A marking $M \in \mathbb{N}^m$ is a dead marking iff the two statements hold:

(i) $\mathcal{S}_\emptyset = \{p \in P \mid M(p) = 0\}$ is an unmarked siphon of N;

(ii) $\forall\, t \in T$, ${}^\bullet t \cap \mathcal{S}_\emptyset \neq \emptyset$.

Proof. (if) It immediately follows from (ii) and the definition of enabled transitions. Condition (i) and (ii) together imply that every transition has at least one empty input place, thus no transition is enabled.

(only if) Condition (i) follows from Lemma 12 while condition (ii) follows from the fact that dead transitions must have in M at least an empty input place, that by definition belongs to $\mathcal{S}_\emptyset$. □

On the basis of the above lemmas, we can finally state the main result.

Theorem 14. Given a structurally bounded net $N = (P, T, Pre, Post)$ with m places, a marking $M \in \mathbb{N}^m$ is a deadlock marking if and only if there exists a vector $\vec{s} \in \{0,1\}^m$ such that the following set of linear equations is satisfied:

$$\mathcal{D}(N) := \begin{cases} K_1 \cdot Pre^T \cdot \vec{s} \geq Post^T \cdot \vec{s} & (a) \\ K_2 \cdot \vec{s} + M \leq K_2 \cdot \vec{1}_m & (b) \\ \vec{s} + M \geq \vec{1}_m & (c) \\ Pre^T \cdot \vec{s} \geq \vec{1} & (d) \\ M \in \mathbb{N}^m & (e) \\ \vec{s} \in \{0,1\}^m & (f) \end{cases} \tag{12}$$

Proof. A marking M is dead if and only if there exists a siphon $\mathcal{S}_\emptyset$ as defined in Lemma 13.

The characteristic vector $\vec{s}$ must satisfy (a) by Lemma 10 and (b) by Lemma 11. Furthermore, by definition of $\mathcal{S}_\emptyset$, if a place p is empty at marking M then it must belong to the siphon, and this is imposed by (c). Finally, (d) states that for any transition t there exists at least one input place that is empty at M, and that consequently belongs to siphon $\mathcal{S}_\emptyset$. □

By virtue of the linear characterization above, we define the set of blocking markings of net N as:

$$\mathcal{M}_b(N) = \{M \;\mid\; \exists\, \vec{s} \in \{0,1\}^m : (M, \vec{s}) \text{ is a solution of } \mathcal{D}(N)\}. \tag{13}$$

Finally, we present a technical result that will be used in the following.

Proposition 15. Given a net $N = (P, T, Pre, Post)$, and a subset of transitions $T' \subsetneq T$, let $N' \prec_{T'} N$ be the $T'-$induced subnet of N. Then $\mathcal{D}(N) \subseteq \mathcal{D}(N')$, or equivalently $\mathcal{M}_b(N) \subseteq \mathcal{M}_b(N')$.

Proof. Let us define $n' = |T'|$ and $n = |T| > n'$. Then it is easy to see that constraints (12).a and (12).d in $\mathcal{D}(N)$ are each composed by n inequalities, i.e., the corresponding n' inequalities in $\mathcal{D}(N')$ plus additional ones. This proves the statement. □

6 Recovery and estimate update after net time-out

Let us suppose that, although we have no exact information on the timing structure of the net, we can be sure that the net is blocked if a sufficiently long time has elapsed without observing any event occurrence. Such is the case if we know that all transition delays are such that $\delta(t) \leq \Delta_{\max}$, $\forall t \in T$. If a time greater than $\Delta_{\max}$ elapses without observing any firing, we say that *the net has timed out*.

Proposition 16. Assume that the net $N = (P, T, Pre, Post)$ controlled with the control pattern $f(\cdot, \mathcal{C})$ has timed out. Let us define $T' = \{t \in T \mid f(t, \mathcal{C}) = 1\}$ as the subset of T containing the transitions enabled by the controller, and let $N' \prec_{T'} N$ be the $T'-$induced subnet of N. Then the actual (unknown) marking M of the controlled net N is a deadlock marking for the uncontrolled net N', i.e., it belongs to $\mathcal{C}' = \mathcal{C} \cap \mathcal{M}_b(N')$.

Proof. The transitions blocked from the controller can be removed from the net N without changing its behavior. The resulting net N' is an autonomous net for which the results of Lemma 13 and Theorem 14 apply. □

We now propose an automatic approach that tries to exploit the information that the net has timed out to recover from this blocking condition and improve the estimate. Of course this procedure may be effective only if the deadlock has been caused by the incomplete information about the actual marking originated by the presence of the observer in the closed loop.

6.1 Deadlock recovery

The deadlock recovery procedure we propose consists in recomputing the control pattern using a new IPP that adds to the constraints in (2) some additional constraint to capture the fact that the actual (unknown) marking M belongs to $\mathcal{M}_b(N')$ for the net N' defined in Proposition 16.

Algorithm 17 (Control Pattern Updating After Net Time-Out)**.** Given a net $N = (P, T, Pre, Post)$ controlled using an observer, let μ and B be the current value of estimate and bound, and define $\mathcal{C} = \mathcal{M}(\mu, B)$. Assume that the computed control pattern $f(\cdot, \mathcal{C})$ has led the net to a time-out. We can update the control pattern using the following procedure.

1. Let $i = 0$ and define $f_0(\cdot) \stackrel{\text{def}}{=} f(\cdot, \mathcal{C})$ as the initial control pattern.
2. Let $T_i = \{t \in T \mid f_i(t) = 1\}$ be the set of transitions enabled by the current control pattern, and let $N_i \prec_{T_i} N$ be the net obtained by N removing all transitions not in T_i.
3. Update the control pattern to $f_{i+1} = g(f_i)$, where
$$g(f_i) \stackrel{\text{def}}{=} f(\cdot, \mathcal{C} \cap \mathcal{M}_b(N_i)). \tag{14}$$
4. If $f_{i+1} = f_i$ THEN exit: the deadlock recovery procedure has failed.
5. Wait until
   (a) EITHER a transition fires and THEN exit: the net has recovered from the deadlock
   (b) OR a new net time-out occurs and THEN let $i = i+1$ and go to 2. ■

Note that the operator $g : \{0,1\}^n \to \{0,1\}^n$ defined by (14) is a function of f_i because N_i is defined using f_i.

In this algorithm the knowledge that a time-out has occurred is used to restrict the set of consistent markings and construct a new control pattern (step 3) that, as the next proposition shows, is at least as permissive as the previous one. If the new control pattern is still blocking and a new time-out occurs the procedure is repeated until either the net recovers from deadlock, or until we cannot update the control pattern any more and the procedure fails.

We now present some elementary results concerning this algorithm.

Proposition 18. Algorithm 17 has the following properties:

- for all i, the updated control pattern computed at step 3 *is at least as permissive as* the previous one, i.e., $f_{i+1} \geq f_i$;
- the algorithm terminates in a finite number of steps;

- if the algorithm terminates at step 4 with $i = \bar{\imath}$, the final control pattern $f_{\bar{\imath}}$ is a fixed point of the operator g.

Proof. The first statement can be proved by induction. In fact we observe (base step) that, by Proposition 9, $\mathcal{C} \cap \mathcal{M}_b(N_0) \subseteq \mathcal{C}$ implies $f_1 = f(\cdot, \mathcal{C} \cap \mathcal{M}_b(N_0)) \geq f(\cdot, \mathcal{C}) = f_0$. Assume now that $f_i \geq f_{i-1}$ for a given i: we prove (induction step) that the same inequality also holds for $i+1$. In fact, $f_i \geq f_{i-1}$ implies $N_{i-1} \prec_{T_{i-1}} N_i$. Thus $\mathcal{C} \cap \mathcal{M}_b(N_i) \subseteq \mathcal{C} \cap \mathcal{M}_b(N_{i-1})$ by Lemma 15 and this implies, by Proposition 9, that

$$f_{i+1} = f(\cdot, \mathcal{C} \cap \mathcal{M}_b(N_i)) \geq f(\cdot, \mathcal{C} \cap \mathcal{M}_b(N_{i-1})) = f_i.$$

The second statement follows from the fact that each time the loop in the algorithm is repeated, either $f_{i+1} = f_i$ (and in this case the algorithm terminates), or, by the previous statement, $f_{i+1} \gneq f_i$ and eventually the maximally permissive control that enables all transitions is reached in a number of steps less or equal to $|T|$.

The third statement follows trivially from the fact that if the algorithm terminates at step 4, then $f_{\bar{\imath}} = f_{\bar{\imath}+1} = g(f_{\bar{\imath}})$. □

6.2 A sufficient condition for deadlock freedom

It is important to characterize those cases in which the procedure outlined in Algorithm 17 is able to recover from a net time-out. Here we consider a particular class of macromarkings, such that the vectors $\vec{v}_j$ are $P-$invariants. In this case, it is possible to show that the set of consistent markings at each step is a subset of the initial macromarking.

Proposition 19. Let the initial macromarking $\mathcal{V}(V, \vec{b})$ be such that $V^T C = \vec{0}$, i.e., each column $\vec{v}_j$ of V is a P-invariant. Then, for all observed words w, $\mathcal{C}(w) \subseteq \mathcal{C}(\varepsilon) \equiv \mathcal{V}(V, \vec{b})$.

Proof. First note that for all observed words w, $V^T \mu_w + B_w = \vec{b}$, whenever V is a matrix of P-invariants. In fact, by Algorithm 4, each time a new transition fires we have $V^T \mu_{wt} + B_{wt} = V^T \left[\mu'_{wt} + C(\cdot, t)\right] + \left[B_w - V^T(\mu'_{wt} - \mu_w)\right] = V^T \mu_w + B_w + V^T C(\cdot, t) = V^T \mu_w + B_w$, while initially, $V^T \mu_\varepsilon + B_\varepsilon = \vec{b}$. Furthermore, $\mu_w \geq \vec{0} = \mu_\varepsilon$, thus for all observed words w, the set of w–consistent markings is $\mathcal{C}(w) = \{M \in \mathbb{N}^n \mid M \geq \mu_w, V^T M = \vec{b}\} \subseteq \{M \in \mathbb{N}^n \mid V^T M = \vec{b}\} = \mathcal{C}(\varepsilon)$. □

We use the previous result to give a sufficient condition to ensure that the controlled net will never time out.

Theorem 20. Consider a net N with initial macromarking $\mathcal{V}(V, \vec{b})$ such that $V^T C = 0$, and controlled using Algorithm 8. Let $T_0 = \{t \in T \mid f(t, \mathcal{C}(\varepsilon)) = 1\}$ be the set of

transitions enabled by the initial control pattern and let $N_0 \prec_{T_0} N$ be the T_0-induced subnet of N.

Then the closed loop system will never reach a blocking state, i.e., the net will never time out, if the following constraint set

$$\begin{cases} V^T M = \vec{b} \\ M \in \mathcal{M}_b(N_0) \end{cases} \tag{15}$$

is not admissible, i.e., if it does not admit any solution for $M \in \mathbb{N}^m$.

Proof. First note that when the net is initialized, the set of consistent markings coincides with the initial macromarking, i.e., $\mathcal{C}(\varepsilon) = \mathcal{V}(V, \vec{b}) = \{M \in \mathbb{N}^m \mid V^T M = \vec{b}\}$. If the constraint set (15) does not admit a feasible solution, the net is never blocked when the control pattern $f(\cdot, \mathcal{C}(\varepsilon))$ is applied, regardless of the initial marking $M \in \mathcal{V}(V, \vec{b})$.

After a word w has been observed, the set of consistent marking is $\mathcal{C}(w) \subseteq \mathcal{C}(\varepsilon)$ (by Proposition 19) while the actual marking still belongs to $\mathcal{V}(V, \vec{b})$, being V a matrix of P-invariants. Thus by Proposition 9 it holds that $f(\cdot, \mathcal{C}(w)) \geq f(\cdot, \mathcal{C}(\varepsilon))$, and regardless of the current marking the controlled net is not blocked. □

We finally extend the previous result, giving a sufficient condition to ensure that, even if a time-out may occur, Algorithm 17 will always successfully recover the net from a deadlock.

Consider a net N with set of consistent markings $\mathcal{C}$. Assume that Algorithm 17 is invoked but at step 5 we always execute step 5.b, until the algorithm stops at step 4 with $f_{i+1} = f_i$: this is the maximally permissive control pattern that could be applied if the net always times out when the set of consistent markings is $\mathcal{C}$. A formal definition is the following.

Definition 21. Given a net N controlled with an observer, and a set of consistent states $\mathcal{C}$, let us define $f_0 \stackrel{\text{def}}{=} f(\cdot, \mathcal{C})$ the initial control vector and let $f_{i+1} = g(f_i)$ for $i \geq 0$.

The *maximal control pattern* associated to $\mathcal{C}$ is $f_{\max}(\cdot, \mathcal{C}) \stackrel{\text{def}}{=} \lim_{i \to \infty} f_i$, i.e., it is the fixed point of g reached iterating from f_0. Note that by Proposition 18 part 2, this fixed point is reached in a finite number of steps (less or equal to the cardinality of the set of transitions T). ■

Theorem 22. Consider a net N with initial macromarking $\mathcal{V}(V, \vec{b})$ such that $V^T C = 0$, and controlled with Algorithm 8. Let $T_{\max} = \{t \in T \mid f_{\max}(t, \mathcal{V}(V, \vec{b})) = 1\}$ be the set of transitions enabled by the maximal control pattern associated to the initial consistent set $\mathcal{C}(\varepsilon) = \mathcal{V}(V, \vec{b})$, as defined in the previous proposition. Let $N_{\max} \prec_{T_{\max}} N$ be the $T_{\max}-$induced subnet of N.

1. If a net time-out occurs and the procedure given in Algorithm 17 is applied, the net will always recover from a deadlock if the following constraint set

$$\begin{cases} V^T M = \vec{b} \\ M \in \mathcal{M}_b(N_{\max}) \end{cases} \tag{16}$$

does not admit any admissible solution for $M \in \mathbb{N}^m$.

2. If $N_{\max} = N$ and the controlled net times-out, Algorithm 17 will recover from the deadlock if and only if the current marking M is not a deadlock marking for the open loop net.

Proof. 1) Firstly, observe that if the constraint set (16) does not admit a feasible solution, the time-out procedure is always capable of recovering from an initial deadlock, because eventually the control pattern $f_{\max}(t, \mathcal{V}(V, \vec{b}))$ will be reached and there exists at least an enabled transition regardless of the initial unknown marking $M \in \mathcal{V}(V, \vec{b})$. Secondly, observe that by induction on the iteration step in Algorithm 17, it is immediate to show that $\mathcal{C}' \subseteq \mathcal{C}''$ implies $f_{\max}(t, \mathcal{C}') \geq f_{\max}(t, \mathcal{C}'')$. Finally, as in the proof of Theorem 20, the result follows from the fact that after a word w has been observed, the set of consistent markings is $\mathcal{C}(w) \subseteq \mathcal{C}(\varepsilon)$ (by Proposition 19) while the actual marking still belongs to $\mathcal{V}(V, \vec{b})$, being V a matrix of P-invariants.

2) If $N_{\max} = N$, then $f_{\max}$ enables all transitions. Assume that the deadlock recovery procedure fails: eventually the control pattern $f_{\max}$ is reached and the controlled net coincides with the open loop net. □

The second part of the theorem is useful to recognize those cases where no block may be ascribed to the controller-observer: in this case, in fact, the time-out procedure will eventually control enable all transitions. Such a case is discussed in Section 6.4.

6.3 Improving the marking estimate

In this subsection, we discuss the possibility of using the linear algebraic characterization above not only to recover from a block, but to improve the marking estimate as well.

Assume that given an observed word w, a current estimate μ_w and bound B_w, a blocking condition occurs, and that after $\bar{\imath}$ iterations of Algorithm 17 a newly enabled transition t fires. At this point, before the firing of t, the set of consistent markings is $\mathcal{M}(\mu_w, B_w) \cap \mathcal{M}_b(N_{\bar{\imath}})$, using the notation defined in the previous subsection. This set corresponds to the dark area in Figure 4.

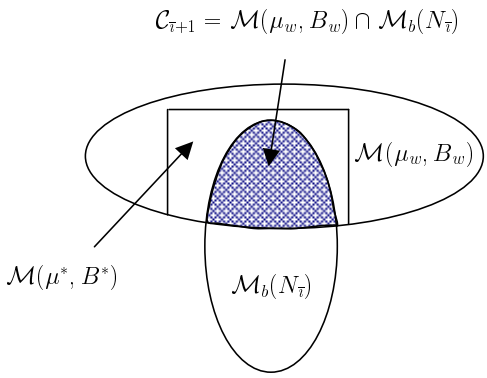

Figure 4: *Generic inclusion relationship among sets* $\mathcal{M}(\mu_w, B_w)$, $\mathcal{M}(\mu^*, B^*)$ *and* $\mathcal{M}_b(N_{\bar{\tau}})$.

We should keep this information when computing the new set of consistent markings $\mathcal{C}(wt)$ after the firing of t. Nevertheless, this would destroy the framework that inspired Algorithm 4, in the sense that the set of consistent markings would loose the structure given in Equation (1).

Thus, we propose the following alternative solution. For each place $p_i \in P$ we solve an IPP of the form:

$$\begin{cases} \min M(p_i) \\ s.t. \\ M \in \mathcal{M}(\mu_w, B_w) \\ M \in \mathcal{M}_b(N_{\bar{\iota}}) \end{cases} \tag{17}$$

Now, we define $\mu^* = [\mu_1^* \ \cdots \ \mu_m^*]^T$ where μ_i^* is the solution of the i–th IPP and let $B^* = B_w - V^T(\mu^* - \mu_w)$ be the corresponding bound. We use μ^* and B^* as new current values of the estimate μ_w and bound B_w, and continue from step 5 of Algorithm 4, computing the updated estimate μ'_{wt}.

This is equivalent to approximate the set of $w-$consistent markings after recovery, with the set

$$\mathcal{M}(\mu^*, B^*) = \{M \in \mathbb{N}^m \mid M \geq \mu^*, V^T \cdot M = V^T \cdot \mu^* + B^*\}. \tag{18}$$

This set is also shown in Figure 4: being $\mathcal{M}(\mu_w, B_w) \cap \mathcal{M}_b(N_{\bar{\iota}}) \subseteq \mathcal{M}(\mu^*, B^*) \subseteq \mathcal{M}(\mu_w, B_w)$ we may be losing information, but nevertheless we can keep on with a linear algebraic characterization of the set of consistent markings in the simple form specified by Equation (1).

6.4 Numerical example

Let us consider again the manufacturing system in Subsection 4.1, where the use of an observer may lead the closed loop system to a blocking condition.

In this subsection we show how the above deadlock recovery procedure may be efficiently applied to the net in Figure 2. Here we assume that although the exact timing structure of the net is unknown to the controller, an upper bound on the transition firing delay $\Delta_{\max} \geq \max_{t \in T} \delta(t)$ is known.

In Figure 5 the top five nodes repeat the net evolution shown in Figure 3.

As already seen in Subsection 4.1 a blocking condition is reached after the firing of the sequence $w = t_1 t_2 t_4 t_3$ at time $now = 11$. When a time $\Delta_{\max}$ has elapsed the net times out and we apply Algorithm 17 to update the control pattern. In particular, we have that the set of transitions enabled by the initial control pattern is $T_0 = T \setminus T_n = \{t_6, t_7\}$, while after the time-out procedure all $T_n = \emptyset$, i.e., all transitions are control enabled. The marking estimate is updated as shown in Figure 5 where the thick arrow has been used to denote that the net has timed out.

When 20 more units of time have elapsed, at time $now = 31 + \Delta_{\max}$, another blocking condition is reached. Thus the net times out at time $now = 31 + 2\,\Delta_{\max}$ and we apply again Algorithm 17. The set of transitions enabled by the controller is now $T_0 = T \setminus \{t_6\}$. In such a case the marking is completely reconstructed as shown in Figure 5, and the net recovers from deadlock.

Finally, let us observe that the initial macromarking considered is such that $V^T C = 0$, thus the assumptions of Theorems 20 and 22 are fulfilled. Therefore, for this net IPP (15) admits feasible solutions and this is in accordance with the fact that the net has timed out.

Moreover, if we compute the maximal control pattern as defined in Definition 21, we find out that $f_{\max}(\cdot, \mathcal{V}(V, \vec{b})) = \vec{1}$, that implies $N_{\max} = N$ according to the notation of Theorem 22. Now, if we consider the set $\{M \in \mathbb{N}^m \mid V^T \cdot M = \vec{b},\ M \in \mathcal{M}_b(N)\}$, we find out that it does not admit any admissible solution for $M \in \mathbb{N}^m$. By Theorem 22 this implies that if a net time-out occurs and we apply the procedure given in Algorithm 17, then the net will always recover from deadlock.

7 Using timing information for state estimation

We now propose a general approach to incorporate available information on the timing structure of the net into the state estimation process. The approach has been firstly pro-

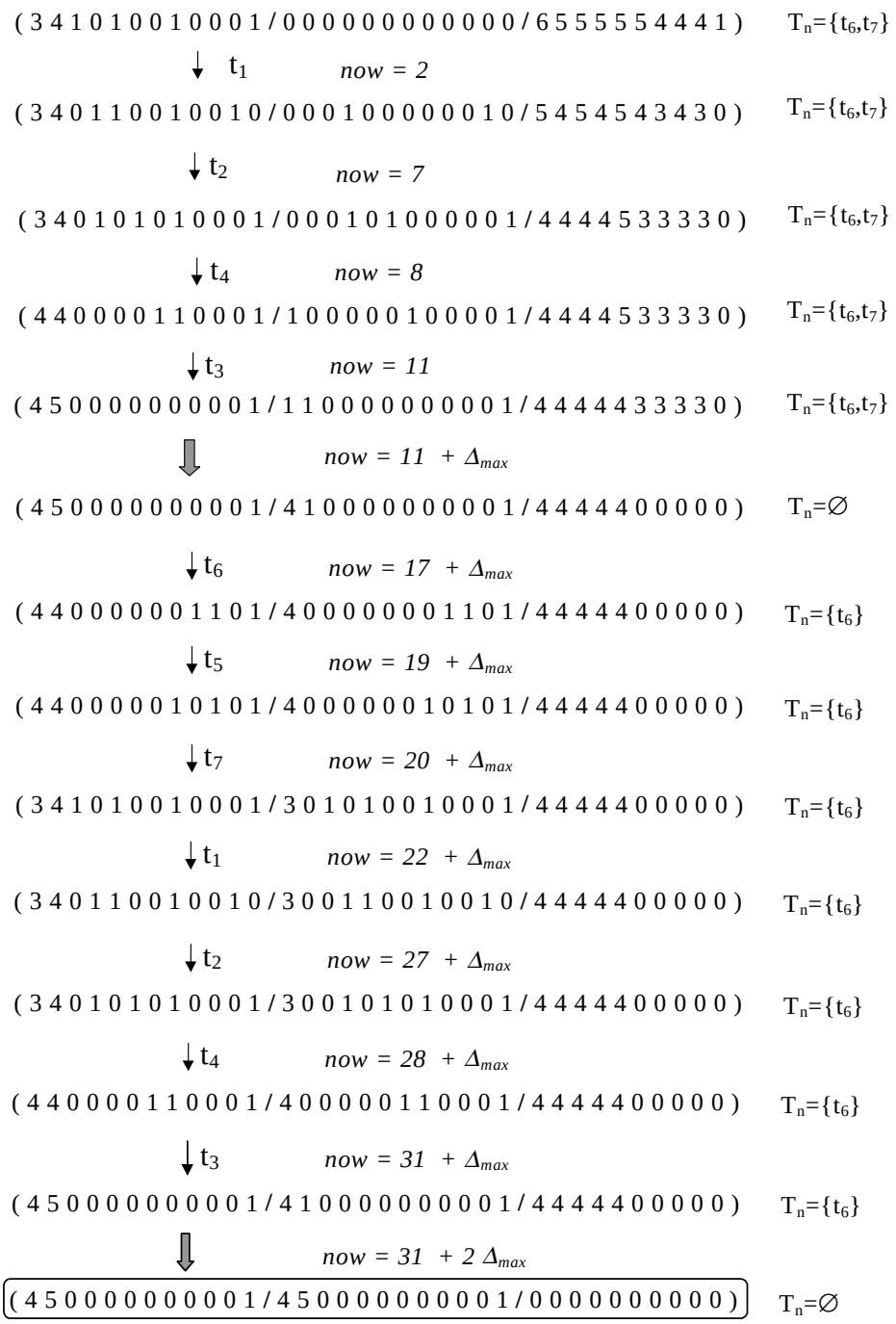

Figure 5: *The evolution of the net in Figure 2 under control when the deadlock recovery procedure proposed in Subsection 6.1 is applied.*

posed by the authors in [3] and is essentially based on the linear algebraic characterization of deadlock markings given by the system of inequalities (12). In particular, the above linear characterization is used to restrict the set of $w-$consistent markings.

7.1 Proposed algorithm

Let us assume that a known delay $\delta(t) : T \to \mathbb{R}$ is associated to each transition; $\delta(t)$ represents the time that must elapse, starting from the time at which the transition t is enabled, until it fires. Assume that we start observing the net at time τ and that transition t is control enabled during the time interval $[\tau, \tau + \delta(t)]$. Moreover, assume that the marking of the input places of t does not increase during the time interval $[\tau, \tau + \delta(t)]$. If at time $now = \tau + \delta(t)$ transition t does not fire, we can be sure that the actual marking M is such that $\neg M[t\rangle$, or equivalently t is not marking enabled: we say that t has *timed out* at time *now*. Note that if the marking of some places in ${}^\bullet t$ has increased during the time interval $[\tau, \tau + \delta(t)]$, we can only conclude that the transition was not marking enabled at time τ, but no conclusion can be drawn on the marking enabling condition of t at time $\tau + \delta(t)$.

The set of timed out transitions is denoted T_{to}.

The procedure that we describe in Algorithm 23 combines the marking estimation algorithm with the deadlock recovery procedure defined in the previous section. It considers two types of events that modify the marking estimate.

- The first type of events occurs when the firing of a transition $\hat{t}$ is detected. In this case the marking estimate μ and bound B are updated following Algorithm 4. In this step the set of timed out transitions T_{to} may eventually be updated, removing from this set all those transitions t such that ${}^\bullet t \cap \hat{t}^\bullet \neq \emptyset$, i.e., those transitions that may have been enabled by the firing of $\hat{t}$.

- The second type of events occurs when a new transition times out. In this case the set of timed out transitions is increased and we know that the actual marking must be such that the net $N_{to} \prec_{T_{to}} N$ is deadlocked, where N_{to} is the subnet of N induced by the set of the timed out transitions. We use this information to compute a new control pattern at least as permissive as the current one. We also update μ and B solving for each place an IPP of the form given by (17).

Algorithm 23 (Control and Estimate Updating After Transition Time-Out)**.** In this algorithm the variable *now* represents the current value of the time. At each instant of time it is possible to partition the set of transitions T into three subsets:

$T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}$ is the set of transitions that are *not control enabled* given the current set of consistent markings.

T_{to} is the set of *control enabled* transitions that have *timed out*. A transition t belongs to this set if during the time interval $[now - \delta(t), now]$ has continuously been control enabled and the marking of all its input places ${}^\bullet t$ has not increased during this same interval.

T_e is the set of those control *enabled* transitions that do not belong to T_{to}.

These are the steps of the algorithm.

1. Let $\mu = \mu_{w0}$ and $B = B_{w0}$ be the initial estimate and bound, and let $\mathcal{C} = \mathcal{M}(\mu_{w0}, B_{w0})$ be the initial set of consistent markings.
2. Compute for all transitions $t \in T$ the control pattern $f(t, \mathcal{C})$ and let $T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}$, $T_{to} = \emptyset$, $T_e = T \setminus T_n$.
3. Set for all transitions $t \in T_e$ the current clock value to $\omega(t) = \delta(t)$.
4. Let $\delta = \min\{\omega(t) \mid t \in T_e\}$ the time-out to wait in step 6.
5. Let $\tau = now$ and $f_{old}(t) = f(t, \mathcal{C})$ (keeps track of the previous pattern).
6. Wait until
   (a) EITHER a transition $\hat{t}$ fires and THEN go to 7
   (b) OR $now = \tau + \delta$ and THEN go to 8.

   Note that if one event of type (a) and one event of type (b) occur simultaneously, then condition 6.a takes priority and the next active step will be 7.
7. Activate the observer update procedure after the firing of $\hat{t}$.
   (a) Update the estimate to μ' with $\mu'(p) = \max\{\mu(p), Pre(p, \hat{t})\}$.
   (b) Let the current estimate and bound be $\mu = \mu' + C(\cdot, \hat{t})$ and $B = B - V^T \cdot (\mu' - \mu)$.
   (c) Let the current set of consistent markings be $\mathcal{C} = \mathcal{M}(\mu, B)$.

(d) Compute for all $t \in T$ the control pattern $f(t, \mathcal{C})$ and let

$$T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\},$$
$$T_{to} = T_{to} \setminus \{t \in T_{to} \mid {}^\bullet t \cap \hat{t}\ {}^\bullet \neq \emptyset\},$$
$$T_e = \{t \in T \mid f(t, \mathcal{C}) = 1,\ t \notin T_{to}\}.$$

(e) Update the clocks of enabled transitions.

IF a transition $t \in T_e$ satisfies at least one of the following three conditions

i. $f_{old}(t) = 0$ {newly control enabled}

ii. ${}^\bullet t \cap \hat{t}^\bullet \neq \emptyset$ {may have become marking enabled by the firing of $\hat{t}$}

iii. $t = \hat{t}$ {it is the transition that has just fired}

THEN $\omega(t) = \delta(t)$ {reset the clock}
ELSE $\omega(t) = \omega(t) - (now - \tau)$.

(f) Go to 4.

8. Activate the time-out procedure.

(a) Let $T_{to} = T_{to} \cup \{t \in T_e \mid \omega(t) = \delta\}$.

(b) Let $N_{to} \prec_{T_{to}} N$ be the $T_{to}-$induced subnet N.

(c) Compute for all transitions $t \in T$ the control pattern $f(t, \mathcal{C} \cap \mathcal{M}_b(N_{to})$ and let

$$T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\},$$
$$T_e = \{t \in T \mid f(t, \mathcal{C}) = 1,\ t \notin T_{to}\}.$$

(d) Improve the previous estimate μ. This simply requires the solution of m linear integer programming problems (IPP), one for each place $p_i \in P$:

$$\begin{cases} \min M(p_i) \\ s.t. \\ M \in \mathcal{M}(\mu, B) \\ M \in \mathcal{M}_b(N_{to}) \end{cases} \tag{19}$$

Now, let $\mu^* = [\mu_1^* \ \cdots \ \mu_m^*]^T$, where μ_i^* is the solution of the i--th IPP, and let $B^* = B - V^T(\mu^* - \mu)$.

(e) Update the estimate and bound to $\mu = \mu^*$ and $B = B^*$, and compute the new set of consistent markings $\mathcal{C} = \mathcal{M}(\mu, B)$.

(f) If $T_e = \emptyset$ exit (the net is deadlocked and the time-out procedure fails to recover), else go to 4. ■

7.2 Numerical example

Let us consider again the manufacturing system in Subsection 4.1 and let us apply Algorithm 23 to compute the closed loop behavior of the net system in Figure 2. The resulting evolution is represented in the reachability graph in Figure 6 where the thick arrow now denotes a time-out and is labeled by the corresponding set T_{to}.

At step 1 we define the initial estimate and bound. At step 2 we compute for all transitions $t \in T$ the control pattern $f(t, \mathcal{C})$ and set $T_n = \{t_6, t_7\}$, $T_{to} = \emptyset$ and $T_e = T \setminus T_n$. In fact, the firing of t_6 may potentially violate specification (b), while the firing of t_7 may potentially violate specification (a). Then, we set up the clock value of each transition in T_e to its time delay. Given the actual delays, the time-out to wait before either applying the observer update procedure or the deadlock recovery procedure, is $\delta = 1$.

At time $now = 1$, no transition fires and t_4 times out. Thus the time-out procedure is activated (step 8). This first implies the updating of $T_{to} = \emptyset$ to $T_{to} = \{t_4\}$. Then, we define the net N_{to} obtained from N removing all transitions not in T_{to}. For all $t \in T$ we compute the new control pattern $f(t, \mathcal{C})$ according to step 8.c and we update the transition partitioning. In particular, we find out that both t_6 and t_7 are still disabled by the controller, thus $T_n = \{t_6, t_7\}$, while $T_e = T \setminus (T_n \cup T_{to})$. Now, by solving $m = 12$ IPP we compute the new marking estimate and bound and go back to step 4 of the algorithm. In such a case, we find out that the updated marking estimate and bound are coincident with the previous ones. We compute the new value δ and, as in the previous step, it holds that $\delta = 1$.

At time $now = 2$, when one more time unit has elapsed, both conditions 6.a and 6.b are simultaneously satisfied because t_1 fires and t_5 times out. Condition 6.a takes priority and transition t_1 fires. The observer update procedure is applied. We update the estimate and bound as shown in Figure 6, while the control pattern keeps the same for all transitions $t \in T$. Note that the firing of t_1 increases the token content of place p_4 that is an input place for t_4: thus t_4 is removed from the set T_{to} at step 7.d. We compute the new value δ and it holds that $\delta = 0$ because t_5 is ready to time out.

Then, always at time $now = 2$, the time-out procedure is activated for t_5. This enables us to improve the marking estimate as shown in Figure 6 and also to make transition t_6 control enabled. More precisely, at time $now = 2$, after the TTO procedure has been applied, it holds that $T_n = \{t_7\}$, $T_{to} = \{t_5\}$ and $T_e = T \setminus (T_n \cup T_{to})$. Once again, at step 4, we find out that $\delta = 1$.

At time $now = 2$, after one more time unit has elapsed, no transition fires. Therefore, the time-out procedure is invoked with $T_{to} = \{t_3, t_4, t_5\}$, and so on.

As it can be seen in Figure 6, at the end of this evolution path, at time $now = 14$, the

marking is completely reconstructed and no further deadlock may occur.

To conclude we may observe that when Algorithm 23 is applied, the closed loop net recovers from the deadlock after 14 time units. On the contrary, when we apply the procedure presented in the previous section, that is invoked only when the net has timed out, the net recovers from the deadlock after more that 43 units of time.

7.3 Linear relaxation of integer programming

A drawback of the proposed procedure is that it requires to solve at each step an integer programming problem to compute the control pattern: in some cases this may hinder the implementation of the approach on on-line controllers. This problem may be partially solved by simply relaxing the integer programming problems we consider into linear ones.

Assume that in IPP (2) the constraints $M, M' \in \mathbb{N}^m$ are relaxed into $M, M' \in (\mathbb{R}_0^+)^m$. This yields a larger set of consistent markings $\mathcal{C}_R(w) \supseteq \mathcal{C}(w)$, i.e., we have a *relaxed observer* (R-observer) that is possibly less accurate than the previously defined observer. By Proposition 9, the control pattern computed using the R-observer is possibly suboptimal, in the sense that it is less permissive than or at most as permissive as the one computed using the observer. Note, however, that the control pattern computed using the R-observer is certainly safe, i.e., it ensures that the control specifications are never violated.

Similarly, if in IPP (12) the constraints $M \in \mathbb{N}^m$ and $\vec{s} \in \{0,1\}^m$ are relaxed into $M \in (\mathbb{R}_0^+)^m$ and $\vec{s} \in [0,1]^m$, this yields a larger set of deadlock markings. In this case the recovery procedures of Algorithm 17 and Algorithm 23 can still be applied but the computed control patterns are, again, possibly suboptimal.

Thus, whenever necessary the control designer may take advantage of the linear relaxation trade-off that allows one to obtain a possibly suboptimal but computationally efficient solution technique.

As a final remark, it may also be possible to combine these techniques using linear programming for the on-line computation of the control patterns, and using integer programming only when applying the net time-out procedure.

As an example, in the case of the Petri net system already considered in Subsections 4.1, 6.4 and 7.2, one may verify that the on-line computation of the control patterns using the linear relaxation of IPP (2) always yield optimal solutions. However, when a net time-out occurs, the linear relaxation is not optimal: the maximal permissive control pattern computed using the linear relaxation of IPP (12) disables $\{t_7\}$ and because of this the deadlock recovery procedure may not work.

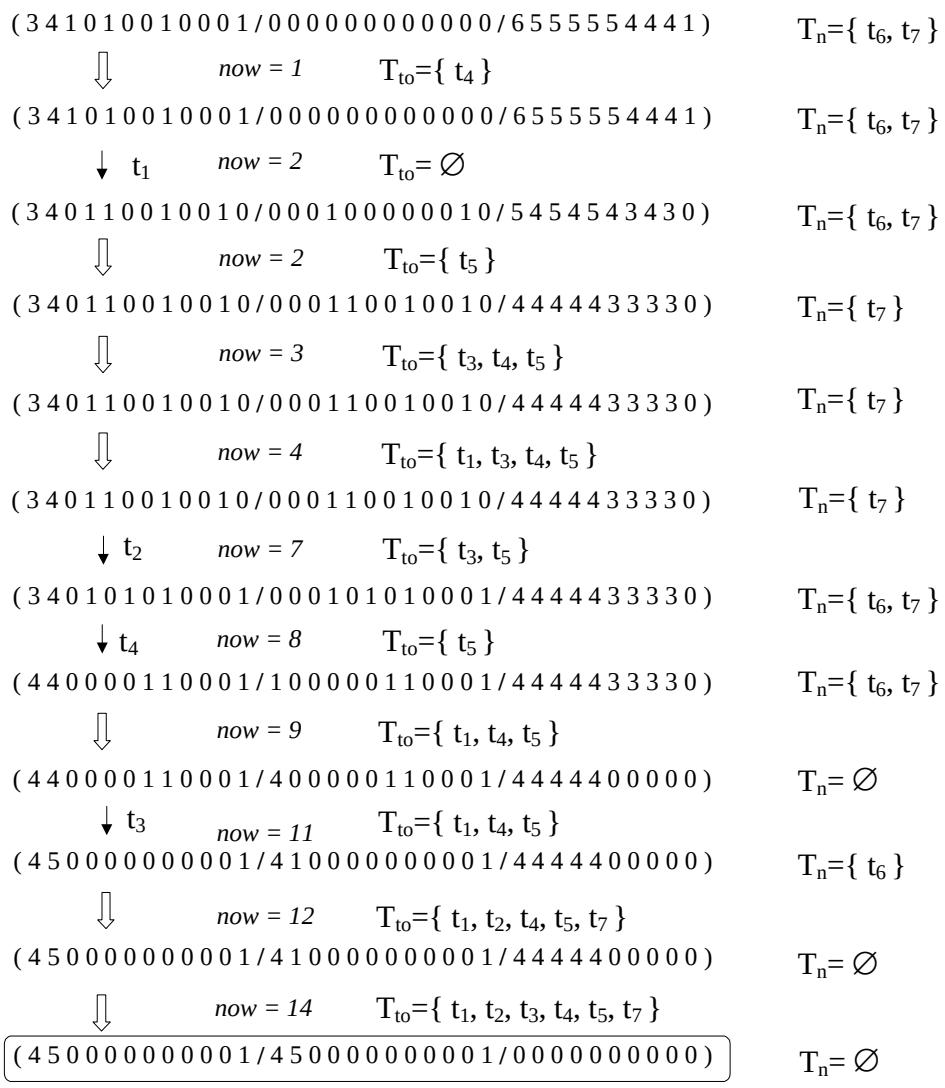

Figure 6: *The evolution of the net in Figure 2 under control when the deadlock recovery procedure using timing information is applied.*

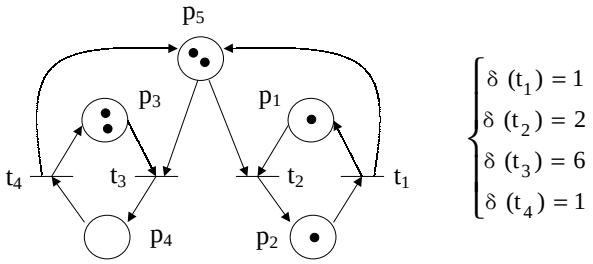

Figure 7: *An example showing how the knowledge of the timing structure may be used to solve partial deadlocks.*

7.4 Properties of the algorithm with transition time-out

The knowledge of the timing structure of a net leads to the possibility of using Algorithm 23 for marking estimate and control. In the following we will call this procedure TTO (transition time-out). Next example shows that the TTO procedure may be able to recover from partial deadlocks: in such a case the NTO procedure (i.e., Algorithm 17) is useless because it is never invoked.

Example 24. Let us consider the net system in Figure 7 with $m = 5$ places and $n = 4$ transitions. There exist 3 circuits, each one corresponding to a P-invariant. If the initial marking is that shown in Figure 7 we have: $\mathcal{V}(V, \vec{b}) = \{M \in \mathbb{N}^5 \mid M_1 + M_2 = 2; M_3 + M_4 = 2; M_2 + M_4 + M_5 = 3\}$. Moreover, we assume that the controller must enforce one specification: $M_1 \leq 1$.

Let us first consider the case in which no information on the timing structure is available. In such a case the net never times out and the behavior is that shown in Figure 8.a where we can observe that a partial deadlock occurs. In fact, transition t_2 may initially fire, but in the sequel only t_3 and t_4 may alternately fire. On the contrary, t_1 is always disabled by the controller because the marking of p_1 has not been reconstructed and its firing may potentially violate the specification.

Now, let us assume that the timing structure is known and the TTO procedure is applied. In such a case the partial deadlock can be solved and the net evolution is that shown in Figure 8.b. At first transition t_1 is disabled by the controller, i.e., $T_n = \{t_1\}$, and $\delta = \delta(t_4) = 1$. Thus, at time $now = 1$ since no transition fires, the TTO procedure is invoked. The set T_n keeps the same but the marking estimate is improved. In particular, we reconstruct the marking of places p_3 and p_4. After one more unit of time transition t_2 fires: once again we improve the marking estimate but we still have that $T_n = \{t_1\}$. On the contrary, at time $now = 4$ the TTO procedure is applied and all transitions become control enabled. Note that at this step, when we update the marking estimate,

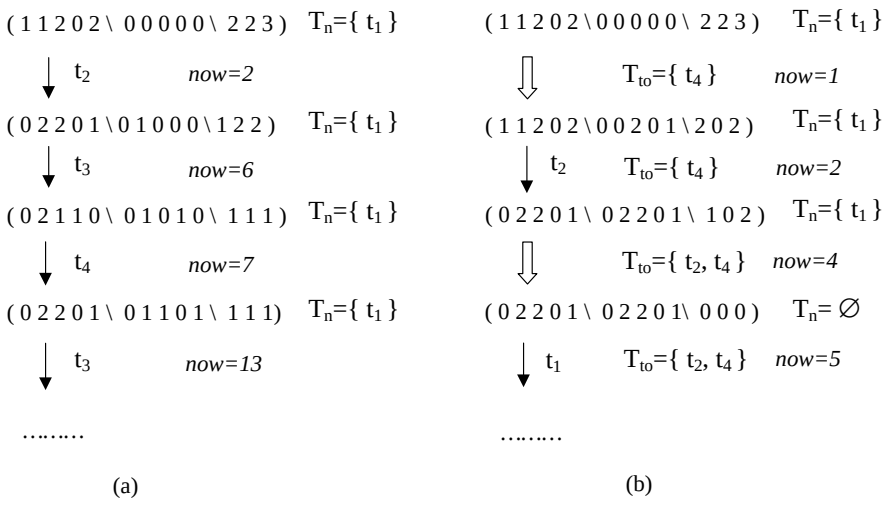

Figure 8: *The behavior of the controlled net in Figure 7: (a) no information on the timing structure is available, (b) the timing structure is known and the TTO procedure is used.*

we completely reconstruct the actual marking of the net. ■

As a final remark, we present a result showing that with respect to total deadlocks the two procedure have the same power.

Theorem 25. Let us consider a net system $\langle N, M\rangle$ controlled with observer with current set of consistent markings $\mathcal{C}$. The net system is deadlocked when controlled with the TTO procedure if and only if it is deadlocked when controlled with the NTO.

Proof. (if) First we show that if the net deadlocks using the NTO procedure it also deadlocks using the TTO procedure.

It the net deadlocks using the NTO procedure, Algorithm 17 is invoked but we always execute step 5.b, until the algorithm stops at step 4 with the maximal control pattern $f_{\max} = f_{\max}(\cdot, \mathcal{C})$ and no transition enabled by it may fire.

Now, let us assume that $\langle N, M\rangle$ is controlled using Algorithm 23 and let $\bar{f}_0 \leq \bar{f}_1 \leq \bar{f}_2 \leq \cdots$ be the series of control patterns computed by repeatedly iterating on step 8. To prove the statement of the theorem, we have to demonstrate that for all $k \geq 0$

$$\bar{f}_k \leq f_{\max}. \tag{20}$$

To prove this, we first observe that the function $g(f)$ is monotone. In fact, given two control patterns f' and f'' with $f' \leq f''$, then $N_{f'} \preceq N_{f''}$. This implies that $\mathcal{M}_b(N_{f'}) \supseteq \mathcal{M}_b(N_{f''}) \Rightarrow g(f') \leq g(f'')$.

Let us define $\bar{T}_i$ as the set of transitions control enabled by $\bar{f}_i$ and $\bar{N}_i$ the corresponding induced subnet. When the updated control vector $\bar{f}_{i+1}$ is computed, only a subset $T_{to,i} \subseteq$

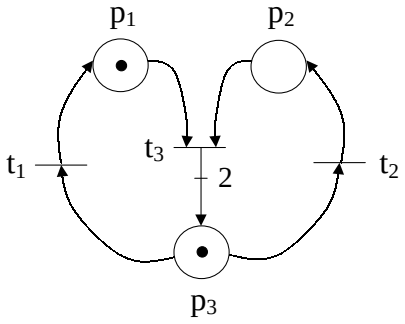

Figure 9: *An example showing that the TTO procedure may fail to recover from a marking induced deadlock.*

$\bar{T}_i$ of these transitions has timed out and, if we define $N_{to,i}$ the corresponding induced subnet, by Proposition 9 we have that

$$\bar{f}_{i+1} = f(\cdot, \mathcal{C} \cap \mathcal{M}_b(N_{to,i})) \leq f(\cdot, \mathcal{C} \cap \mathcal{M}_b(\bar{N}_i)) = g(\bar{f}_i). \tag{21}$$

We now prove by induction that equation (20) holds for all values of k. In fact $\bar{f}_0 = f(\cdot, \mathcal{C}) \leq f_{\max}$. Assume now $\bar{f}_k \leq f_{\max}$. Then by (21) and by the monotonicity property, we also have that $\bar{f}_{k+1} \leq g(\bar{f}_k) \leq g(f_{\max}) = f_{\max}$.

(only if) We show that if the net does not deadlock using the NTO procedure then it also does not deadlock using the TTO procedure. This result trivially follows from the fact that if we wait a sufficiently long time, then all transitions eventually time out and $\lim_{i\to\infty} \bar{f}_i = f_{\max}$. □

To conclude, we present a very simple example showing that in some cases the TTO procedure does not preserve the liveness of the system.

Example 26. Let us consider the Petri net system in Figure 9 whose unknown initial marking is $M_0 = [1\ 0\ 1]^T$. The initial macromarking is $\mathcal{V}(V, \vec{b}) = \{M \in \mathbb{N}^3 \mid M_1 + M_2 = 1,\ M_3 = 1\}$ and let us assume that the controller must enforce the specifications: $M_1 \leq 1$ and $M_2 \leq 1$.

The controlled net is live if the controller exactly knows the actual marking. As an example, given the marking in figure, the controller disables t_1 and enables transition t_2.

On the contrary, regardless of the timing structure, the controlled net is dead if the observer is included in the closed loop. In fact, from the macromarking equation $M_1 + M_2 = 1$ it is impossible to know whether p_1 or p_2 is initially marked, hence both t_1 and t_2 must be control disabled. When transition t_3 times out, no additional information on the location of the token in the set $\{p_1, p_2\}$ can be inferred and the recovery procedure fails. ■

8 Conclusions

In this paper we have dealt with the problem of enforcing a set of GMEC on a timed Petri net by a state feedback control under the assumption that the system state is not measurable but can only be estimated. We showed by means of an example that the use of an estimate instead of the actual marking, may lead to a deadlock even if the controlled system is live. In the case that the net system is structurally bounded, we propose an algorithm that accelerates the state estimation and helps us to detect the observer induced deadlock. We first consider the case in which no information on the timing structure is known, then we show how the procedure may be modified when the delays associated to transitions are known. We also prove that this information may also be used to improve the marking estimate and to recover the net from partial deadlocks.

References

[1] K. Barkaoui, A. Chaoui, B. Zouari, "Supervisory control of discrete event systems using structure theory of Petri nets ," *1997 IEEE Int. Conf. on Systems, Man and Cybernetics* (Orlando, Florida), pp. 3750-3755, Oct 1997.

[2] F. Basile, P. Chiacchio, A. Giua, C. Seatzu, "Deadlock recovery of controlled Petri net models using observers," *8th IEEE Int. Conf. on Emerging Technologies and Factory Automation* (Antibes, France), pp. 441–449, Oct 2001.

[3] F. Basile, A. Giua, C. Seatzu, "Petri net control using event observers and timing information," *41st IEEE 2002 Conference on Decision and Control* (Las Vegas, Nevada), pp. 787-792, Dec 2002.

[4] F. Chu, X. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 6, pp. 793–804, 1997.

[5] J. Ezpeleta, J.M. Colom, J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems", *IEEE Trans. on Robotics & Automation*, Vol. 11, No. 2, pp. 173–184, 1995.

[6] A. Giua, F. DiCesare. M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," *Proc. 1992 IEEE Int. Conf. on Systems, Man, and Cybernetics* (Chicago, Illinois), pp. 974–979, Oct 1992.

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

[8] L. E. Holloway, B. H. Krogh, A. Giua, "A survey of Petri net methods for controlled discrete event systems", *Discrete Event Systems*, Vol. 7, pp. 151-190, 1997.

[9] Y. Li, W.M. Wonham, "Controllability and observability in the state-feedback control of discrete-event systems," *Proc. 27th Conf. on Decision and Control* (Austin, Texas), pp. 203–207, Dec 1988.

[10] Y. Li, W.M. Wonham, "Control of vector discrete-event systems — part I: the base model," *IEEE Trans. on Automatic Control*, Vol. 38, No. 8, pp. 1215–1227, 1993.

[11] Y. Li, W.M. Wonham, "Control of vector discrete-event systems — part II: controller synthesis," *IEEE Trans. on Automatic Control*, Vol. 39, No. 3, pp. 512–531, 1994.

[12] J.O. Moody, P.J. Antsaklis, "Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions," *IEEE Trans. Automatic Control*, Vol. 45, No. 3, pp. 462–476, 2000.

[13] T. Murata, "Petri nets: properties, analysis and applications," *Proc. IEEE*, Vol. Proc. 77, N. 4, pp. 541–580, 1989.

[14] J. Park, S.A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. on Automatic Control*, Vol. 46, No. 10, pp. 1572–1583, 2001.

[15] Di Cesare, F., G. Harhalakis, J.M. Proth, M. Silva and F.B. Vernadat, *Practice of Petri nets in manufacturing*, Chapman and Hall, 1993.

[16] A. Ramírez–Treviño, I. Rivera–Rangel, E. López–Mellado, "Observer design for discrete event systems modeled by interpreted Petri nets," *2000 IEEE Int. Conf. on Robotics and Automation*, pp. 2871–2876, Apr 2000.

[17] W. Reisig, *Petri nets. An introduction*, Springer–Verlag, 1982.

[18] M. Silva, J.M. Colom, "On the computation of structural synchronic invariants in P/T nets," in *Advances in Petri Nets 1988*, Springer–Verlag, 1989.

[19] S. Takai, T. Ushio, S. Kodama, "Static-state feedback control of discrete-event systems under partial observation," *IEEE Trans. on Automatic Control*, Vol. 40, No. 11, pp. 1950–1955, 1995.

[20] K. Yamalidou, J.O. Moody, M.D. Lemmon, P.J. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, Vol. 32, No. 1, 1996.

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