Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Petri net control using event observers and timing information

Alessandro Giua, Carla Seatzu
Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{giua,seatzu}@diee.unica.it

Francesco Basile
Dip. Ing. dell'Informazione e Ing. Elettrica, Università di Salerno
Via Ponte don Melillo, 84084 Fisciano (Salerno), Italy
fbasile@unisa.it

Abstract

The use of an observer in Petri net control may significantly reduce the performance of the closed-loop system because of the incomplete knowledge of the net marking. In this paper we present an algorithm that uses the information on the timing structure associated to the net to improve the marking estimate. This procedure may be invoked not only when the controlled system has entered a blocking condition (as in a previous work) but also whenever a transition has not fired for a time larger than its expected delay. The algorithm requires solving a number of integer linear programming problems and helps us to detect partial deadlocks and to accelerate the convergence of the marking estimation procedure.

1 Introduction

In this paper we deal with the issue of controlling a timed Petri net whose marking cannot be measured. The state-feedback control of discrete event systems with incomplete information has already been discussed in the literature [6, 7, 11, 13]. In particular, we assume that the net structure is completely known while the initial marking is only known to belong to a "macromarking", i.e., we know the token contents of subsets of places but not the exact token distribution.

In previous works [6, 7] it was 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 and error bound. 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 and error computation. In particular, the set $\mathcal{C}$ of markings consistent with an observed word, 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.

In [6, 7] we have 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 specifications that limit the weighted sum of markings in subsets of places called generalized mutual exclusion constraints (GMEC).

Clearly, the presence of an observer in the feedback loop, i.e., 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, the controller may disable transitions whose firing is perfectly legal, and because of this it may be the case that the controlled system is blocking. This problem was tackled in [2] where we showed that the set of deadlock markings of a structurally bounded P/T net can be characterized as the integer solutions of a linear constraint set. We assumed that if no transition firing occurs within a reasonable amount of time in a controlled system, a deadlock has occurred and a deadlock recovery procedure is invoked. In such a procedure the additional information that the controlled net is deadlocked is used to reduce the set of consistent markings $\mathcal{C}$, thus obtaining a better estimate of the actual marking. In fact, given a net N where T' is the subset of transitions enabled by the controller, if no transition can fire, then the actual marking M must also belong to the set $\mathcal{M}_b(N')$ of blocking markings for the net N' obtained from N removing all transitions not in T'. This smaller set of consistent markings $\mathcal{C} \cap \mathcal{M}_b(N')$ can also be characterized by a linear constraint set. A smaller set of consistent markings may allow us to compute a less restrictive control pattern and to recover from a deadlock created by the presence of an observer in the feedback loop.

In this paper we extend this result and 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 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. In fact, when a transition t has been control enabled for a time longer than its delay without firing (we say that it has timed-out), then we know that the actual marking does not enable t. If T_{to} is the set of transitions that have timed out, repeating the previous reasoning we can be sure that the actual marking M must also belong to the set of blocking markings for the net N_{to} obtained from N removing all transitions not in T_{to}.

This algorithm not only allows the controller to recover from a total deadlock (as in [2]) but it allows one to detect partial deadlocks as well, and in general it improves and accelerates the convergence of the marking estimation procedure.

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 $X^\bullet$

0-7803-7516-5/02/$17.00 ©2002 IEEE

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 transition t is enabled at M if $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t)$. We write $M\ [w\rangle\ M'$ 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 w_0 the sequence of null length.

A marking M is *reachable* in N from M_0 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 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 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.

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 be thought as obtained from N 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 $\delta(t) : T \rightarrow \mathbb{R}_0^+$, 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.

Finally, we conclude this section recalling a linear algebraic characterization of deadlock markings derived in [2] that will be used in the paper. Such a characterization is valid for ordinary and structurally bounded Petri nets. Note that similar linear characterizations have been independently proposed in [1, 3, 10].

Theorem 2 ([2]) *Given a structurally bounded net N 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:*

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

where $K_1 = \max_{t \in T} Post^T(\cdot, t) \cdot \vec{1}$ and K_2 is any positive integer greater or equal to the maximum structural bound of p, for any $p \in P$. ■

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

$$\mathcal{M}_b(N) = \{M \mid \exists\, \vec{s} \in \{0,1\}^m : (M, \vec{s}) \in \mathcal{D}(N)\}. \quad (2)$$

3 Marking estimation with macromarkings

In this paper we assume that partial information about the initial marking is available. In particular, we assume that the initial marking is given in the form of a *macromarking*.

Definition 3 ([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 4 ([6]) After the word w has been observed we define the set of w–consistent markings as

$$\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \exists M_0 \in \mathcal{V}(V, \vec{b}),\ M_0[w\rangle M\}.$$

i.e., as the set of all markings in which the system may be given the observed behaviour and the initial marking. ■

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 estimate μ_w and bound B_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 $\mathcal{V}(V, \vec{b})$.

Algorithm 5 ([6]) *Marking Estimation with Event Observation and Initial Macromarking*

```
1. Let the initial estimate be $\mu_{w_0} = \vec{0}_m$.
2. Let the initial bound be $B_{w_0} = \vec{b}$.
3. Let the current observed word be $w = w_0$.
4. Wait until $t$ fires.
5. Update the estimate $\mu_w$ to $\mu'_{wt}$ 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. ■

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

Definition 6 Given a net with initial macromarking $\mathcal{V}(V, \vec{b})$, a current estimate μ and bound B computed by Algorithm 5, we define the *set of* (μ, B)*-consistent markings*

$$\mathcal{M}(\mu, B) = \{M \in \mathbb{N}^n \mid M \geq \mu,\ V^T \cdot M = V^T \cdot \mu + B\}. \tag{3}$$

■

Theorem 7 ([6]) Given a net with initial macromarking $\mathcal{V}(V, \vec{b})$, an observed word w, and the corresponding estimated marking μ and bound B computed by Algorithm 5, the set of $w-$*consistent markings* coincides with the set of (μ, B)-consistent markings, i.e., $\mathcal{C}(w) = \mathcal{M}(\mu, B)$. ■

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 behaviour [7].

We make several assumptions that are briefly discussed here.

- The specification is given as a set of forbidden markings $\mathcal{F}$. The set of legal markings is $\mathcal{L} = \mathbb{N}^m - \mathcal{F}$.
- We consider a special type of state specifications called *generalized mutual exclusion constraints* (GMEC) that have been considered by various authors [5, 8, 12].

  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 states

  $$\mathcal{L} = \{M \in \mathbb{N}^m \mid L^T \cdot M \leq \vec{k}\}.$$

- The controller may disable transitions to prevent the plant from entering a forbidden marking, computing a control pattern $f(t, M) : T \times \mathbb{N}^m \rightarrow \{0, 1\}$. If $f(t, M) = 0$ then t is disabled by the controller at M.
- All transitions are controllable, i.e., can be disabled by the controller.

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 $f(t, \mathcal{C}) : T \times 2^{\mathbb{N}^m} \rightarrow \{0, 1\}$ and can be given as follows.

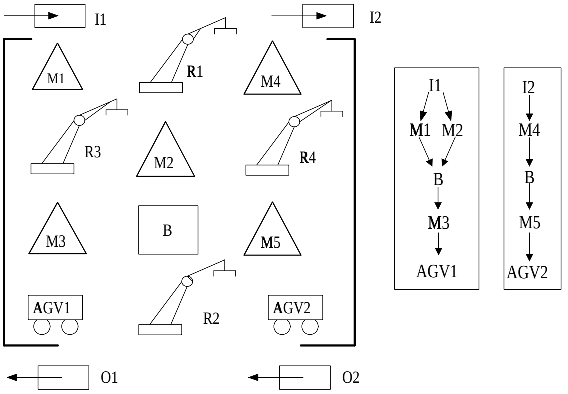


Figure 1: *Layout of the automated manufacturing system.*

Definition 8 (Optimal state feedback 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

1. For all transitions 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

   $$\begin{cases} \max\ \vec{l}_j^T \cdot M' & \\ \text{s.t.} & \\ M \in \mathcal{C} & (a) \\ L^T \cdot M \leq \vec{k} & (b) \\ M \geq Pre(\cdot, t) & (c) \\ M' = M + C(\cdot, t) & (d) \end{cases} \tag{4}$$

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

   $$f(t, \mathcal{C}) = \begin{cases} 0 & \text{if } (\exists j \in J_t) h_j(t) > k_j \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

   the desired control pattern. ■

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 under the assumption that the actual marking is legal, we need not solve IPP (4) for all those constraints such $\vec{l}_j^T \cdot C(\cdot, t) \leq 0$, because they may never be violated by the firing of t.

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

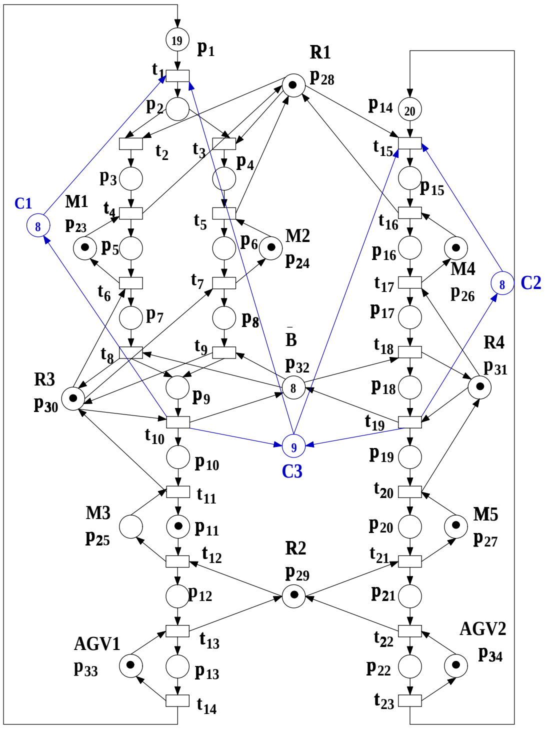


Figure 2: *Petri net model of the manufacturing system in figure 1.*

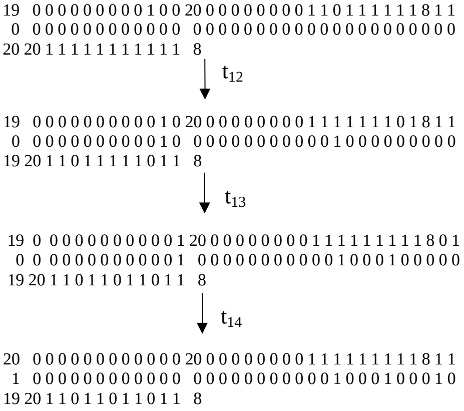
4.1 A manufacturing example

We now apply the above methodology to a classical automated manufacturing system whose layout is shown in figure 1 and whose Petri net model is shown in figure 2 (places $C1$, $C2$, $C3$ and all connected arcs should be ignored at first). This system is similar to the one described in [14].

The plant consists of five machines (M1 to M5), four robots (R1 to R4), a finite capacity buffer B, two inputs of raw parts (I1 and I2) of type1 and type2 respectively, two AGV systems (AGV1 and AGV2), and finally two outputs (O1 and O2) for the processed parts. The plant produces two different types of products from two types of raw materials. An unlimited source of raw parts is assumed. It is supposed that there are 20 pallets for each type of product.

The Petri net model in figure 2 belongs to a special class of Petri nets called S^3PR [4]. This net has $m = 34$ places and $n = 23$ transitions. The marking of place p_{32}, the co-buffer, represents the number of free buffer slots, while the marking of places p_9 and p_{18} represent respectively the number of type1 and type2 parts present in the buffer. There exist 14 circuits, each corresponding to a P-invariant. If we assume that the initial marking of the net is that in figure 2, we have (here to avoid a heavy notation we denote as M_i the marking of place p_i) $\sum_{i=1}^{13} M_i = 20$, $\sum_{i=14}^{22} M_i = 20$, $M_5 + M_{23} = 1$, $M_6 + M_{24} = 1$, $M_{11} + M_{25} = 1$, $M_{16} + M_{26} = 1$, $M_{20} + M_{27} = 1$, $M_{13} + M_{33} = 1$, $M_{22} + M_{34} = 1$, $M_3 + M_4 + M_{15} + M_{28} = 1$, $M_{12} + M_{21} + M_{29} = 1$, $M_7 + M_8 + M_{10} + M_{30} = 1$, $M_{17} + M_{19} + M_{31} = 1$, $M_9 + M_{18} + M_{32} = 8$. We assume that the above set of P-invariants coincides with the macromarking, thus $B_{w_0} = \vec{b} = [20\ 20\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 8]^T$.

Note that if the number of P–invariants is too high to be taken into account, we can only ”keep a subset of it”.



```
19  0 0 0 0 0 0 0 0 0 1 0 0 20 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 8 1 1
 0  0 0 0 0 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20 20 1 1 1 1 1 1 1 1 1 1 1   8
                              ↓ t12
19  0 0 0 0 0 0 0 0 0 0 1 0 20 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 8 1 1
 0  0 0 0 0 0 0 0 0 0 0 1 0   0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
19 20 1 1 0 1 1 1 1 1 0 1 1   8
                              ↓ t13
19 0  0 0 0 0 0 0 0 0 0 0 0 1 20 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 8 0 1
 0 0  0 0 0 0 0 0 0 0 0 0 0 1   0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
19 20 1 1 0 1 1 0 1 1 0 1 1   8
                              ↓ t14
20  0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 8 1 1
 1  0 0 0 0 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0
19 20 1 1 0 1 1 0 1 1 0 1 1   8
```

Figure 3: *Reachability graph of the net in figure 2 under control when no deadlock recovery procedure is applied.*

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

$$\left\{ \begin{array}{ll} \sum_{i=2}^{9} M_i \leq 8 & (a) \\ \sum_{i=15}^{18} M_i \leq 8 & (b) \\ \sum_{i=2}^{9} + \sum_{i=15}^{18} M_i \leq 9 & (c) \end{array} \right. \qquad (6)$$

Note that if the initial marking is completely known, the addition of the monitor places $C1$, $C2$ and $C3$ ensures the satisfaction of the linear inequality constraints (6) and the closed loop net is live (this may be easily proved following the procedure in [4]).

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 a deadlock. The closed loop behaviour is that shown in figure 3 where the first line of each node contains the real marking of the net, the second line contains the actual estimate and the third line contains the actual bound.

After the sequence $w = t_{12}t_{13}t_{14}$ has fired, only two transitions t_1 and t_{15} are enabled in the net. The controller prevents the firing of both transitions even if their firing is perfectly legal and the net reaches a deadlock. This is due to the fact that there exists at least one marking in $\mathcal{C}(t_{12}t_{13}t_{14})$ that would produce the violation of one of the controller specifications if either transition t_1 or t_{15} fires. In particular, the firing of t_1 may (potentially) violate specifications (a-c), while the firing of t_{15} may violate specifications (b-c).

5 Control with marking estimation and time–outs

In this section we propose a general approach to exploit available information on the timing structure of the net so as to obtain a better estimate of the set of consistent markings. The approach is essentially based on the linear algebraic characterization of deadlock markings given by the system of inequalities (1).

Let us assume that a known delay $\delta(t) : T \to \mathbb{R}$ is associated to each transition. We say that a transition t has *timed-out* at time *now* if it has been control enabled without firing during the time interval $[now - \delta(t), now]$ and the marking of its input places ${}^{\bullet}t$ has not increased during this interval.

Thus, we can be sure that at time now the actual marking M is such that $\neg M[t\rangle$, or equivalently t is not marking enabled. The set of timed-out transitions is denoted T_{to}.

The procedure that we describe in algorithm 9 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, while the second type of events occurs when a new transition times-out.

Algorithm 9 *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
$$\begin{array}{l} T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}, \\ T_{to} = \emptyset, \\ T_e = \{t \in T \mid f(t, \mathcal{C}) = 1\}. \end{array}$$
3. Set for all $t \in T_e$ the current clock to $\omega(t) = \delta(t)$.
4. Let $\delta = \min\{\omega(t) \mid t \in T_e\}$ the time to wait (step 6).
5. Let $\tau = now$ and $f_{old}(t) = f(t, \mathcal{C})$ (keeps track of the previous control 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.

   If one event of type (a) and one event of type (b) occur simultaneously, then condition 6.a takes priority.
7. Activate the observer update procedure.
   (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 transitions $t \in T$ the control pattern $f(t, \mathcal{C})$ and let
$$\begin{array}{l} T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}, \\ T_{to} = T_{to} \setminus \{t \in T_{to} \mid {}^\bullet t \cap \hat{t}^\bullet \neq \emptyset\}, \\ T_e = \{t \in T \mid f(t, \mathcal{C}) = 1, \; t \notin T_{to}\}. \end{array}$$
   (e) Update the clocks. - For all transitions $t \in T_e$ such that EITHER $f_{old}(t) = 0$ (newly control enabled transitions) OR ${}^\bullet t \cap \hat{t}^\bullet \neq \emptyset$ (transitions who may have become marking enabled by the firing of $\hat{t}$) LET $\omega(t) = \delta(t)$.
   - For all transitions $t \in T_e$ such that $f_{old}(t) = 1$ AND ${}^\bullet t \cap \hat{t}^\bullet = \emptyset$ (transitions who were control enabled before the firing of $\hat{t}$ and that cannot have become marking enabled by the firing of $\hat{t}$) LET $\omega(t) = \omega(t) - (now - \tau)$.
   (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
$$\begin{array}{l} T_n = \{t \in T \mid f(t, \mathcal{C}) = 0\}, \\ T_e = \{t \in T \mid f(t, \mathcal{C}) = 1, \; t \notin T_{to}\}. \end{array}$$
   (d) Improve the previous estimate μ. This simply requires the solution of m linear integer programming problems (IPP), one for each place $p_i \in P$:
$$\left\{ \begin{array}{l} \min M(p_i) \\ s.t. \\ M \in \mathcal{M}(\mu, B) \\ M \in \mathcal{M}_b(N_{to}) \end{array} \right. \qquad (7)$$
   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 from the deadlock), else goto 4. ■

The main idea behind the algorithm is the following. If T_{to} is the set of transitions that have timed out at time now we can be sure that the actual marking M must also belong to the set of blocking markings for the net N_{to} obtained from N removing all transitions not in T_{to}. Thus in step 8.(c) we can compute a (possibly) less restrictive control pattern using as set of consistent markings $\mathcal{C} \cap \mathcal{M}_b(N_{to})$.

This set, even if defined by a set of linear inequalities — namely, the constraint set of IPP (7) — is not in the simple form given by eq. (3) that is required in the following step of the algorithm. Thus at step 8.(d) we approximate it with a set of the form given by eq. (3) computing new estimates and bounds.

5.1 Numerical example

Let us consider again the manufacturing system in subsection 4.1 and assume that delay times are associated to transitions. In particular, we assume: $\delta(t) = 5$ for all $t \in T \setminus \{t_{12}, t_{13}, t_{14}, t_{21}, t_{22}, t_{23}\}$, $\delta(t) = 1$ for $t \in \{t_{13}, t_{14}, t_{22}, t_{23}\}$, and $\delta(t) = 2$ for $t \in \{t_{12}, t_{21}\}$.

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 $T_n = \{t_1, t_{15}\}$, $T_{to} = \emptyset$ and $T_e = T \setminus T_n$. 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 $\delta = 1$. In this case, after one time unit has elapsed, no transition fires. In fact, none among all transitions $t \in T_e$ such that $\omega(t) = 1$, namely t_{13}, t_{14}, t_{22} and t_{23}, may actually fire even if their firing is allowed by the controller. Thus, the time-out procedure is activated (step 8).

This first implies the updating of $T_{to} = \emptyset$ to $T_{to} = \{t_{13}, t_{14}, t_{22}, t_{23}\}$. 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_1 and t_{15} are still disabled by the controller, thus $T_n = \{t_1, t_{15}\}$, while $T_e = T \setminus (T_n \cup T_{to})$. Now, by solving $m = 34$ IPP we update the previous marking estimate and bound and go back to step 4 of the algorithm. Numerical values are reported in figure 4 where rectangular border has been used to highlight that the time–out procedure has been applied but no transition has fired.

At step 4 we compute the new value of δ and we find out that as in the previous step $\delta = 1$. At this point, when one more time unit has elapsed, transition t_{12} fires and the observer update procedure is applied. We update the estimate and bound as shown in figure 4, while the control pattern keeps the same for all transitions $t \in T$. Moreover, being $^\bullet t_{13} \cap t_{12}^\bullet \neq \emptyset$, the sets T_{to} and consequently T_e should be updated. In particular, we have that $T_{to} = \{t_{14}, t_{22}, t_{23}\}$ and $T_e = T \setminus (T_n \cup T_{to})$, where T_n is the same as in the previous step.

Then, after one more time unit t_{13} fires, and after another time unit t_{14} fires as well. The resulting marking estimate and bound are those reported in figure 4, respectively in the fourth and fifth nodes.

As in the previous steps, $\delta = 1$ but no transition fires thus the time-out procedure is invoked. The new control pattern is computed and all transitions become control enabled. The marking estimate is also updated as in the sixth node in figure 4. Now, both t_1 or t_{15} may fire.

6 Conclusions

We have considered the problem of enforcing a set of GMEC on a Petri net system by a state feedback control and under the hypothesis that the state is not measurable but can only be estimated. An algorithm that accelerates the state estimation based on the knowledge of the timing structure of net has been presented. This algorithm may also allow the controlled net to recover from a deadlock.

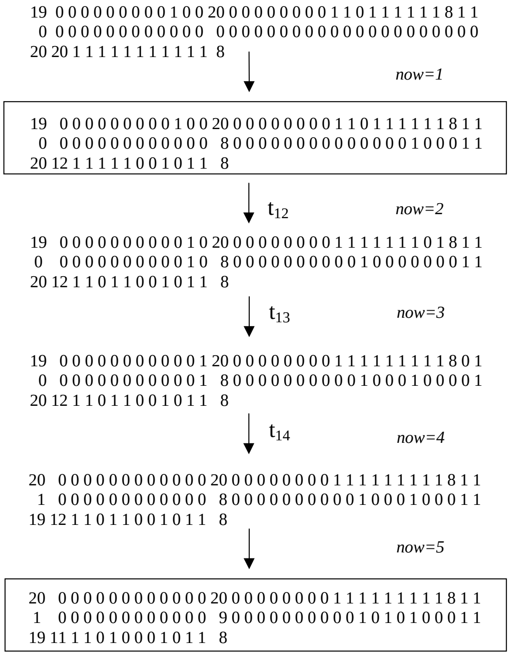


Figure 4: *The evolution of the net in figure 2 under control when Algorithm 9 is applied.*

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

[4] 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.

[5] 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.

[6] A. Giua, "Petri net state estimators based on event observation," *Proc. 36th Int. Conf. on Decision and Control* (San Diego, California), pp. 4086–4091, December 1997.

[7] A. Giua, C. Seatzu, "Observability of place/transition nets," *IEEE Trans. on Automatic Control*, Vol. 47, 2002.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] M.C. Zhou, F. DiCesare, *Petri net synthesis for discrete event control of manufacturing systems*. Kluwer, 1993.