Continuous Petri Nets: Expressive Power and Decidability Issues

Laura Recalde[2], Serge Haddad[1], Manuel Silva[2] *

[1] LAMSADE-CNRS UMR 7024, University Paris-Dauphine, France
E-mail: haddad@lamsade.dauphine.fr

[2] GISED, University of Zaragoza, Spain
E-mail: {lrecalde | silva}@unizar.es

Abstract. State explosion is a fundamental problem in the analysis and synthesis of discrete event systems. Continuous Petri nets can be seen as a relaxation of discrete models. The expected gains are twofold: improvements in comlexity and in decidability. This paper concentrates on the study of decidability issues. In the case of autonomous nets it is proved that properties like reachability, liveness or deadlock-freeness remain decidable. When time is introduced in the model (using an infinite server semantics) decidability of these properties is lost, since continuous timed Petri nets are able to simulate Turing machines.

1 Introduction

State explosion problems represent a main drawback in the study of heavily loaded discrete event dynamic systems, modeled for example as Petri nets. Trying to alleviate these problems, different relaxations have been used, in particular, continuous Petri nets. The expected gains are twofold: improvements in computability and in decidability. An example of computability gain is the study of reachability, that under very general conditions (consistent continuous nets with all the transitions fireable) can be computed in polynomial time [8]. Another property that is improved, now for timed nets, is the computation of an initial marking that maximizes a linear function of the throughput (production), the steady-state marking (work in process) and the initial marking (investment). In continuous equal conflict nets (the weighted version of free choice nets) this initial marking can be computed using a linear programming problem [11]. This paper concentrates on the study of decidability issues. The autonomous model is studied first, and it is proved that properties like reachability, liveness or deadlock-freeness, that are decidable in discrete Petri net systems [4], are also decidable in continuous Petri net systems, as it was expected.

In discrete Petri nets, time has been introduced in many different ways. Here we will concentrate on one of them: a delay is associated to transitions according to a distribution function. Discrete PNs with general probability distribution functions associated to transitions (which includes deterministic timing)

* This work was partially supported by project CICYT and FEDER DPI2003-06376.

are equivalent to Turing machines [3]. However, this is based on the existence of transitions that fire in zero time (immediate transitions), that introduce a notion of priority. In continuous timed Petri nets, these transitions will not be considered, so the model is more restrictive in this sense.

In [2] decidability problems of hybrid Petri nets were studied, proving that reachability was decidable. The model was hybrid i.e., it combined a discrete with a continuous part. Regarding the interpretation, the discrete part was untimed, while the dynamics of the continuous part was based on finite server semantics. In some sense, the model used here is simpler (all the transitions are continuous), but more complex in the dynamics used for the marking evolution (the so called infinite server semantics). Under infinite servers semantics the evolution of the marking can be represented as a set of constant parameters linear differential equations. The switching among them is associated to minimum operators. In [6], an alternative model was defined (timed differentiable Petri nets), and it was shown that it could model a Turing machine. Here it is proved that these two models are equivalent. Moreover, self-loop arcs can be avoided in the model. That is, the modelling power of pure continuous Petri nets under infinite server semantics is equivalent to that of Turing machines.

The structure of the paper is as follows: in Section 2 autonomous and timed continuous Petri nets are introduced, illustrating the kind of behavior they can model with several examples. Decidability of properties like reachability, liveness or deadlock-freeness of the autonomous model is studied in Section 3. Section 4 deals with the timed model, proving that timed continuous Petri nets and timed differentiable Petri nets are in fact equivalent. This is used in Section 5 to deduce that timed continuous Petri nets have Turing machine modelling power.

2 Continuous Petri Nets

2.1 Autonomous Continuous Petri Nets

We assume that the reader is familiar with Petri nets (PNs) (for notation we use the standard one, see for instance [10]).

The structure $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post}\rangle$ of *(autonomous) continuous Petri nets* (ACPN) is the same as the structure of discrete PNs. That is, P is a finite set of places, T is a finite set of transitions with $P \cap T = \emptyset$, $\mathbf{Pre}$ and $\mathbf{Post}$ are $|P| \times |T|$ sized, natural valued, *pre- and post- incidence matrices*. The usual PN system, $\langle \mathcal{N}, \mathbf{m_0}\rangle$, will be said to be *discrete* so as to distinguish it from a continuous PN system, in which $\mathbf{m_0} \in (\mathbb{R}_{\geq 0})^{|P|}$. The main difference between both formalisms is in the evolution rule, since in continuous PNs firing is not restricted to be done in integer amounts. As a consequence the marking is not forced to be integer. More precisely, a transition t is *enabled* at $\mathbf{m}$ iff for every $p \in {}^\bullet t$, $\mathbf{m}[p] > 0$, and its *enabling degree* is $\mathrm{enab}(t, \mathbf{m}) = \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p, t]\}$. The firing of t in a certain amount $\alpha \leq \mathrm{enab}(t, \mathbf{m})$ leads to a new marking $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}[P, t]$, where $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the token-flow matrix.

As in discrete systems, right and left natural annullers of the token flow matrix are called T- and P-semiflows, respectively. When $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$, $\mathbf{y} > \mathbf{0}$ the

net is said to be *conservative*, and when $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$, $\mathbf{x} > \mathbf{0}$ the net is said to be *consistent*. A set of places Θ is a *trap* iff $\Theta^\bullet \subseteq {}^\bullet\Theta$. Similarly, a set of places Σ is a *siphon* iff ${}^\bullet\Sigma \subseteq \Sigma^\bullet$. The support of a vector $\mathbf{v} \geq \mathbf{0}$ will be denoted as $\|\mathbf{v}\|$ and represents the set of positive elements of $\mathbf{v}$.

In continuous PNs the reachability concept is not so immediate as in discrete nets. For example, in a continuous net it may happen that the marking of a place can be done smaller and smaller, but never reaches 0. This idea of getting as close as desired to a marking, even if it is never reached with a finite firing sequence leads in [9] to the definition of limit reachability, further refined in [8].

Definition 1. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system. Two reachability concepts are defined:*

- $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \{\ \mathbf{m} \in (\mathbb{R}_{\geq 0})^{|P|} \mid$ *a finite fireable sequence* $\sigma = \alpha_1 t_{a_1} \dots \alpha_k t_{a_k}$ *exists such that* $\mathbf{m_0} \xrightarrow{\alpha_1 t_{a_1}} \mathbf{m}_1 \xrightarrow{\alpha_2 t_{a_2}} \dots \xrightarrow{\alpha_k t_{a_k}} \mathbf{m}_k = \mathbf{m}$ *with* $t_i \in T$ *and* $\alpha_i \in \mathbb{R}_{\geq 0}\}$.
- $\text{lim-}\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \{\mathbf{m} \in (\mathbb{R}_{\geq 0})^{|P|} |$ *a sequence of reachable markings* $\{\mathbf{m}_i\}_{i \geq 1}$ *exists verifying* $\mathbf{m_0} \xrightarrow{\sigma_1} \mathbf{m}_1 \xrightarrow{\sigma_2} \mathbf{m}_2 \cdots \mathbf{m}_{i-1} \xrightarrow{\sigma_i} \mathbf{m}_i \cdots$ *and* $\lim_{i \to \infty} \mathbf{m}_i = \mathbf{m}\}$.

The set of reachable (and lim-reachable) markings in continuous PNs satisfies some properties that do not hold for discrete nets. For example, the reachability set (and the lim-reachability set) of a continuous system is a convex set [9].

Many basic properties of discrete PNs can be extended to continuous PNs.

Definition 2. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system.*

- $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is* (lim-) *deadlock-free iff for every* $\mathbf{m} \in$ (lim-) $\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ *there exists* $t \in T$ *such that* $\mathrm{enab}(t, \mathbf{m}) > 0$.
- $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is* (lim-) *live iff for every* $\mathbf{m} \in$ (lim-) $\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ *and for every* $t \in T$ *there exist* $\mathbf{m}' \in$ (lim-) $\mathrm{RS}(\mathcal{N}, \mathbf{m})$ *such that* $\mathrm{enab}(t, \mathbf{m}') > 0$.
- $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is* (lim-) *reversible iff for every* $\mathbf{m} \in$ (lim-) $\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ *then* $\mathbf{m_0} \in$ (lim-) $\mathrm{RS}(\mathcal{N}, \mathbf{m})$.

Since continuous PN are a relaxation of discrete PN, for those properties based on universal (existential) quantifiers the continuous PN will provide sufficient (necessary) conditions. For example, if the continuous PN is bounded, so will be the discrete PN. For a marking to be reachable in the discrete model, reachability in the continuous one must be guaranteed. However, for those properties formulated interleaving universal and existential quantifiers the analysis of the continuous PN may not provide information about the behavior of its discrete counterpart. For example, liveness of the continuous PN is neither necessary nor sufficient for liveness of the discrete model [9].

Besides the situation in which the properties of the discrete and the continuous net are not related, it also happens that some properties of discrete PN cannot be observed in continuous systems, as mutex relationships. Moreover, the distinction between two properties may be lost in the continuous model. For example, under broad conditions, lim-liveness and lim-reversibility are equivalent.

Indeed, reformulating Theorem 21 in [8], the following result can be proved:

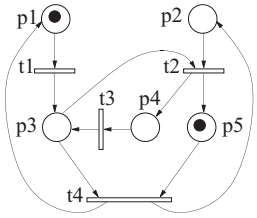

Fig. 1. This system is not reversible as discrete, or as continuous with finite number of firings, but it is lim-reversible

Theorem 1. $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is consistent and* lim*-live iff it is* lim*-reversible and every transition is fireable at least once.*

However, liveness and consistency are not sufficient conditions for reversibility in discrete systems, and neither are for continuous net systems if finite firing sequences are considered. For example, the system in Fig. 1 is consistent and live as discrete, however once t_1 has fired it is impossible to get back to the initial marking. In the continuous net system, to go back to the initial marking from $\mathbf{m} = [0, 0, 1, 0, 1]$, an infinite sequence $\frac{1}{2}t_4\frac{1}{2}t_2, \frac{1}{2}t_3, \frac{1}{4}t_4, \frac{1}{4}t_2, \frac{1}{4}t_3, \ldots \frac{1}{2^k}t_4, \frac{1}{2^k}t_2, \frac{1}{2^k}t_3, \ldots$ has to be fired.

2.2 Timed Continuous Petri Nets

A simple and interesting way to introduce time in discrete PNs is to assume that all the transitions are timed with exponential probability distribution functions. For the timing interpretation of continuous PNs we will use a first order (or deterministic) approximation of the discrete case, assuming that the delays associated to the firing of transitions can be approximated by their mean values. These mean delays will be assumed to be positive, i.e., immediate transitions are not allowed.

Definition 3. *A* Timed Continuous Petri Net (TCPN) *is a continuous PN together with a vector* $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$.

A TCPN with an initial marking $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ *will be denoted a TCPN system.*

Since it is an interpretation of the autonomous net, the evolution of a TCPN has to fulfill the state equation: $\mathbf{m}(\tau) = \mathbf{m}(0) + \mathbf{C} \cdot \boldsymbol{\sigma}(\tau)$, where $\mathbf{m}$ and σ now depend on τ, the actual time. Deriving, $\dot{\mathbf{m}}(\tau) = \mathbf{C} \cdot \mathbf{f}(\tau)$, where $\mathbf{f}(\mathbf{m}) = \dot{\boldsymbol{\sigma}}(\tau)$ is the flow obtained by firing the transitions. Different semantics have been used to define this flow, the two most important being *infinite server* (or *variable speed*) and *finite server* (or *constant speed*) [1,11]. Here infinite server semantics will be considered.

Like in purely markovian discrete net models, under *infinite server semantics*, the flow through a timed transition t is the product of the speed, $\boldsymbol{\lambda}[t]$, and $\text{enab}(t, \mathbf{m})$, the instantaneous enabling of the transition, i.e.,

$$\mathbf{f}(\mathbf{m})[t] = \boldsymbol{\lambda}[t] \cdot \text{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \{\mathbf{m}[p] / \mathbf{Pre}[p, t]\}.$$

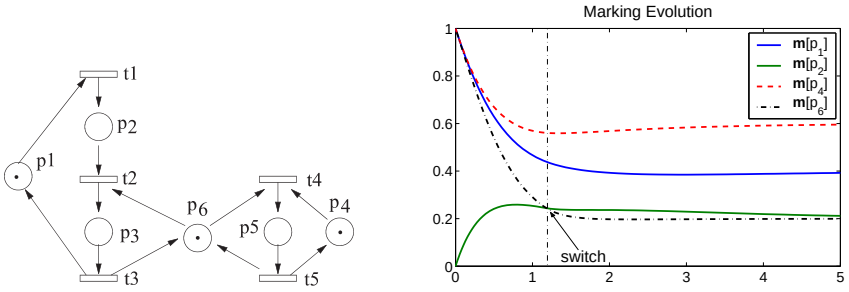

Fig. 2. A TCPN and its marking evolution with $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]$. The system switches only once, and approaches to a steady-state.

For the flow to be well defined, every transition must have at least one input place, hence in the following we will assume $\forall t \in T, |{}^\bullet t| \geq 1$.

Notice that the flow is a piecewise linear function, i.e., the continuous timed model is *technically hybrid*. The change of behavior happens when in a synchronization the place representing the minimum changes. Hence, the switching among the linear systems is given by an internal event. A system without synchronizations (i.e., for every t $|{}^\bullet t| = 1$) would be linear.

Let us introduce the concept of *configurations*: a configuration assigns to a transition one place that for some markings will control its firing rate. Thus the number of configurations is $\Pi_{t \in T} |{}^\bullet t|$. The reachability space can be divided into *regions* according to the configurations. These regions are polyhedrons, and are disjoint, except on the borders. Inside each polyhedron, the evolution of the system is defined by a linear differential equation.

Different kind of behaviors can be modeled with TCPN. In the following we will show some examples to illustrate their modelling power.

Steady-state with finite number of switches. Let us consider the continuous relaxed view of the PN system in Fig. 2 with $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]$. The flows through transitions are given by:

$$
\begin{aligned}
\dot{\mathbf{m}}[p_1] &= \mathbf{f}[t_3] - \mathbf{f}[t_1] = \mathbf{m}[p_3] - \mathbf{m}[p_1] \\
\dot{\mathbf{m}}[p_2] &= \mathbf{f}[t_1] - \mathbf{f}[t_2] = \mathbf{m}[p_1] - 2 \cdot \min(\mathbf{m}[p_2], \mathbf{m}[p_6]) \\
\dot{\mathbf{m}}[p_3] &= \mathbf{f}[t_2] - \mathbf{f}[t_3] = 2 \cdot \min(\mathbf{m}[p_2], \mathbf{m}[p_6]) - \mathbf{m}[p_3] \\
\dot{\mathbf{m}}[p_4] &= \mathbf{f}[t_5] - \mathbf{f}[t_4] = 0.5 \cdot \mathbf{m}[p_5] - \min(\mathbf{m}[p_4], \mathbf{m}[p_6]) \\
\dot{\mathbf{m}}[p_5] &= \mathbf{f}[t_4] - \mathbf{f}[t_5] = \min(\mathbf{m}[p_4], \mathbf{m}[p_6]) - 0.5 \cdot \mathbf{m}[p_5] \\
\dot{\mathbf{m}}[p_6] &= \mathbf{f}[t_3] + \mathbf{f}[t_5] - \mathbf{f}[t_2] - \mathbf{f}[t_4] = \\
&= \mathbf{m}[p_3] + 0.5 \cdot \mathbf{m}[p_5] - 2 \cdot \min(\mathbf{m}[p_2], \mathbf{m}[p_6]) - \min(\mathbf{m}[p_4], \mathbf{m}[p_6])
\end{aligned}
$$

With the initial marking shown in Fig. 2, there is one switch when $\mathbf{m}[p_4] = \mathbf{m}[p_6]$, and then the system approaches its steady-state and never switches again.

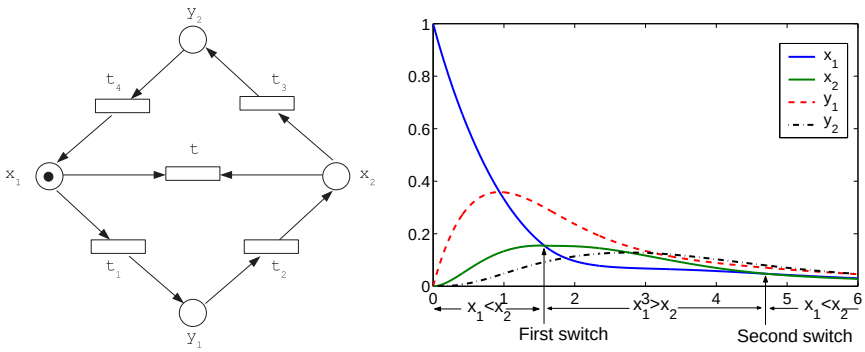

Fig. 3. A TCPN and its marking evolution with $\boldsymbol{\lambda} = [1, 1, 1, 1, 1]$. The system switches an infinite number of times while approaching to a steady-state.

Steady state but infinite switches. Looking at the previous example, one may wonder whether the asymptotical behavior of a net approaching a steady-state can be reduced to the behavior of the set of configurations. This would be the case if switches between configurations occurred only a finite number of times, as happens in the previous example. However, it can be proved that the net system of Fig. 3 approaches to a steady state while infinitely switching between configurations. This net has two regions, namely $\{\mathbf{m} \mid \mathbf{m}[x_1] \geq \mathbf{m}[x_2]\}$ and $\{\mathbf{m} \mid \mathbf{m}[x_1] \leq \mathbf{m}[x_2]\}$. The steady state belongs to both regions, i.e., it is a border point, and the system keeps switching between them.

Periodic behavior without switches. A periodic behavior in which the configuration does not change can also be modeled with TCPN. The ordinary differential equation corresponding to the net system of Fig. 4 is:

$$\begin{aligned}
\dot{\mathbf{m}}[x_1] &= 2a\omega \cdot \min\left\{\frac{\mathbf{m}[x_2]}{2a}, \mathbf{m}[y_1]\right\} - a\omega \cdot \min\left\{\mathbf{m}[x_1], \frac{\mathbf{m}[pk]}{a}\right\} \\
\dot{\mathbf{m}}[x_2] &= a\omega \cdot \min\left\{\mathbf{m}[y_2], \frac{\mathbf{m}[pk]}{a}\right\} - 2a\omega \cdot \min\left\{\mathbf{m}[x_2], \frac{\mathbf{m}[x_1]}{2a}\right\} \\
\dot{\mathbf{m}}[y_1] &= a\omega \cdot \min\left\{\mathbf{m}[x_1], \frac{\mathbf{m}[pk]}{a}\right\} - 2a\omega \cdot \min\left\{\frac{\mathbf{m}[x_2]}{2a}, \mathbf{m}[y_1]\right\} \\
\dot{\mathbf{m}}[y_2] &= 2a\omega \cdot \min\left\{\mathbf{m}[x_2], \frac{\mathbf{m}[x_1]}{2a}\right\} - a\omega \cdot \min\left\{\mathbf{m}[y_2], \frac{\mathbf{m}[pk]}{a}\right\}
\end{aligned} \tag{1}$$

This net has sixteen configurations. However, for $1 \leq a \leq b \leq 2a-1$, the system never switches and always remains in the configuration defined by $\mathbf{m_0}$.

Infinite repetitive switches (not approaching a steady state). The behavior of the system represented by the model in Fig. 5 shows that infinite switches without approaching to a final marking can also be modeled with TCPN. In this net there are two configurations that are commuting indefinitely (observe that sometimes $\mathbf{m}[p_3] < \mathbf{m}[p_4]$, and sometimes $\mathbf{m}[p_4] < \mathbf{m}[p_3]$).

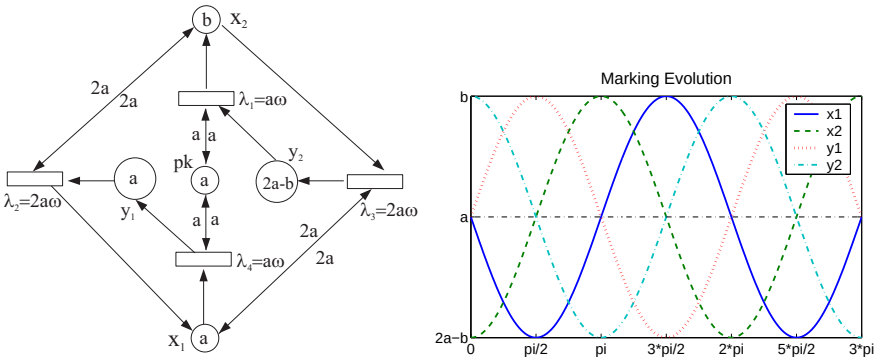

Fig. 4. A TCPN and its marking evolution with $\boldsymbol{\lambda} = [a\omega, 2a\omega, 2a\omega, a\omega]$. The system has a periodic behavior but always remains inside the same configuration.

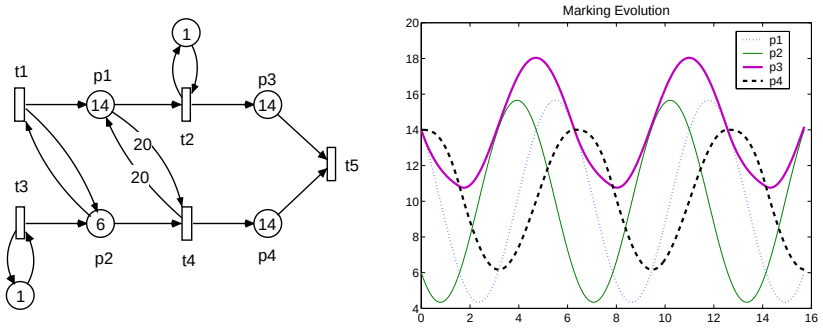

Fig. 5. A TCPN and its marking evolution with $\boldsymbol{\lambda} = [1, 10, 10, 20, 1]$. The system switches indefinitely, and does not approach to a limit marking.

3 Decidability of Basic Properties of Autonomous Continuous Petri Nets

The idea under continuization is that it leads to "easier to analyze" models. For that, we need to ensure that properties can be analyzed at least. This section will be devoted to proving that properties like (lim-)reachability, (lim-)liveness and (lim-)deadlock-freeness are decidable.

In [8], a characterization of reachability and lim-reachability is presented.

Theorem 2. *Let* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *be an ACPN system. Marking* $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ *iff*

1. $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0},\ \boldsymbol{\sigma} \geq \mathbf{0}$
2. $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$
3. *no trap in* $P \setminus \|\mathbf{m}\|$ *intersects with* $\|\mathbf{m_0}\| \cup \|\boldsymbol{\sigma}\|^{\bullet}$

where $FS(\mathcal{N}, \mathbf{m_0})$ *is the set of the supports of sequences fireable from* $\mathbf{m_0}$*, which is a finite set that can be effectively constructed.*

A marking $\mathbf{m} \in \text{lim-RS}(\mathcal{N}, \mathbf{m_0})$ *iff it verifies conditions (1) and (2).*

For the computation of $FS(\mathcal{N}, \mathbf{m_0})$, first add all the combinations of transitions that are enabled at $\mathbf{m_0}$. Then, take one of these sets and fire all the transitions, but in an amount smaller than the enabling degree. This will possibly enable other transitions, so new sets are added to FS. Repeat the procedure till all the sets in FS have been checked.

The only difference between reachability and lim-reachability is on traps, which can be emptied in the limit, but not with a finite sequence. In [8], decidability of reachability is proved. This result can be generalized to lim-reachability.

Corollary 1. *Reachability and* lim-*reachability are decidable for ACPN systems.*

The characterization of (lim-)reachability allows to address also the problem of (lim-)deadlock freeness.

Theorem 3. *For ACPN systems deadlock-freeness and* lim-*deadlock-freeness are decidable.*

Proof. Let us see that checking deadlock-freeness can be reduced to solving a set of linear programming problems.

Let $DP = \{DP_i\}_{i \in I}$ be the set of all the sets of places that have at least one input place per transition. Hence applying Theorem 2, $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$ is a deadlock iff there exist $DP_i \in DP$ and $FS_j \in FS$ such that

$$\mathbf{m_0}[DP_i] + \mathbf{C}[DP_i, FS_j] \cdot \boldsymbol{\sigma}[FS_j] = \mathbf{0} \tag{2}$$

$$\mathbf{m_0}[P \setminus DP_i] + \mathbf{C}[P \setminus DP_i, FS_j] \cdot \boldsymbol{\sigma}[FS_j] > \mathbf{0} \tag{3}$$

$$\boldsymbol{\sigma}[FS_j] > \mathbf{0} \tag{4}$$

$$\text{For every trap } \Theta \subseteq DP_i, (\|\mathbf{m_0}\| \cup FS_j{}^{\bullet}) \cap \Theta = \emptyset \tag{5}$$

Applying the characterization of traps in [12], (5) is equivalent to checking whether the solution of the following linear programming problem is zero,

$$\begin{aligned} &\text{maximize } \varepsilon_{i,j} \\ &\text{subject to: } \mathbf{y} \cdot \mathbf{C}_\Theta \geq \mathbf{0} \\ &\qquad \mathbf{y} \geq \mathbf{0} \\ &\qquad \mathbf{y}[P \setminus DP_i] = \mathbf{0} \\ &\qquad \sum_{p \in \|\mathbf{m_0}\| \cup FS_j{}^{\bullet}} \mathbf{y}[p] \geq \varepsilon_{i,j} \end{aligned} \tag{6}$$

where C_Θ is the token flow matrix of the net $\mathcal{N}_\Theta = \langle P, T, \mathbf{Pre}, \mathbf{Post}_\Theta \rangle$ with $\mathbf{Post}_\Theta[p,t] = 0$ iff $\mathbf{Post}[p,t] = 0$, and $\mathbf{Post}_\Theta[p,t] \geq \sum_{p' \in {}^{\bullet}t} \mathbf{Pre}[p',t]$.

Equations (2), (3) and (4) are equivalent to:

$$\begin{aligned} &\text{maximize } \delta_{i,j} \\ &\text{subject to: } \mathbf{m_0}[DP_i] + \mathbf{C}[DP_i, FS_j] \cdot \boldsymbol{\sigma}[FS_j] = \mathbf{0} \\ &\qquad \mathbf{m_0}[P \setminus DP_i] + \mathbf{C}[P \setminus DP_i, FS_j] \cdot \boldsymbol{\sigma}[FS_j] \geq \delta_{i,j} \cdot \mathbf{1} \\ &\qquad \boldsymbol{\sigma}[FS_j] \geq \delta_{i,j} \cdot \mathbf{1} \end{aligned} \tag{7}$$

Therefore, to prove that the system is deadlock-free, check that for each $DP_i \in DP$ and each $FS_j \in FS$ either (6) has a positive solution or (7) does not have a positive solution.

Regarding lim-deadlock-freeness, the only difference is that there is no restriction with respect to the traps, so clearly it is also decidable. □

Theorem 4. *Liveness and* lim*-liveness are decidable for ACPN systems.*

Proof. Let us study liveness first. For continuous PNs, a transition t is not fireable for any successor of $\mathbf{m}$ iff there exists an empty siphon in $\mathbf{m}$ that contains a place $p \in {}^\bullet t$ (Lemma 11 in [8]). Hence, the system is non live iff there exist $FS_j \in FS$ and a siphon Σ such that

$$\mathbf{m} = \mathbf{m_0} + \mathbf{C}[P, FS_j] \cdot \boldsymbol{\sigma}[FS_j] \tag{8}$$

$$\mathbf{m} \geq \mathbf{0} \tag{9}$$

$$\boldsymbol{\sigma}[FS_j] > \mathbf{0} \tag{10}$$

$$\mathbf{m}[\Sigma] = 0 \tag{11}$$

$$\text{For any trap } \Theta \text{ such that } (\|\mathbf{m_0}\| \cup FS_j{}^\bullet) \cap \Theta = \emptyset, \exists p \in \Theta \text{ with } \mathbf{m}[p] = 0 \tag{12}$$

Let $\{\Sigma_i\}_{i \in I}$ be the set of minimal siphons (i.e., siphons that are not contained in other siphons), and let $\{\Theta_k\}_{k \in K}$ be the set of traps. Then for each $FS_j \in FS$, each trap Θ_k verifying $(\|\mathbf{m_0}\| \cup FS_j{}^\bullet) \cap \Theta_k \neq \emptyset$, and each siphon Σ_i, check whether the following linear programming problem has a positive solution:

$$\begin{aligned} &\text{maximize } \varepsilon_{i,j} \\ &\text{subject to: } \mathbf{m} = \mathbf{m_0} + \mathbf{C}[P, FS_j] \cdot \boldsymbol{\sigma}[FS_j] \\ &\qquad\qquad \mathbf{m} \geq \mathbf{0} \\ &\qquad\qquad \mathbf{m}[\Sigma_i] = 0 \\ &\qquad\qquad \boldsymbol{\sigma}[FS_j] \geq \varepsilon_{i,j} \cdot \mathbf{1} \\ &\qquad\qquad \mathbf{m}[p] = 0 \end{aligned}$$

Regarding lim-liveness, notice that in fact it is equivalent that a transition is not fireable from any reachable marking or from any lim-reachable marking. Then, the lim-liveness problem is analogous, changing the reachability condition, which amounts to forgetting the traps, and removing the last equation ($\mathbf{m}[p] = 0$) in each one of the linear programming problems. □

Combining this result with Theorem 1, the following corollary is obtained.

Corollary 2. lim*-reversibility is decidable in ACPN systems in which all the transitions are fireable.*

4 Timed Continuous Petri Nets and Timed Differentiable Petri nets

The structure and equations of TCPN were somehow inherited from those of discrete Petri nets. Petri nets are based on a production/consumption logic, and

it is the flow of material that is mainly represented. However, these material flow channels can also be used to simulate control flows, and self-loop structures appear. This kind of structures can be used in continuous PNs to "force" behaviors. For example, in the net in Fig. 4, places y_1 and y_2 never define the flow of their output transitions (i.e., never belong to the active configuration), and this has been achieved by a tuning in the self-loop arcs that connect the other places to the transitions. That means that for example t_2 has two input places, but one of them is used to control the speed of the transition, while the other is the one that "suffers" the changes.

Moreover, the information that appears in the structure of the net is redundant. Assume a place-transition-place-transition subnet, and let us denote them as $p_1 - t_1 - p_2 - t_2$. The marking evolution of p_2 due to the output flow does not depend on the weight of the arc (p_2, t_2), and its input flow really depends on the quotient of the weights (p_1, t_1) and (t_1, p_2).

Hence, the notation of TCPN "looks cumbersome", and the evolution rules seem convoluted and counterintuitive in some cases. Trying to clarify the behavior, in [6], a different way of introducing time in a PN structure was presented, in which two different kinds of arcs are used: one to model the *control*, and the other to model the *marking evolution*. This is similar to what is done in Forrester diagrams, in which *control* and *material* flows are kept separate [5].

Definition 4. *A Timed Differentiable Petri Net (TDPN)* $\mathcal{D} = \langle P, T, \mathbf{C}, \mathbf{W} \rangle$ *is defined by:*

- $\langle P, T, \mathbf{C} \rangle$*, a pure PN (thus* $\mathbf{C}$ *is the incidence matrix),*
- $\mathbf{W}$*, the speed control matrix: a mapping from* $P \times T$ *to* $\mathbb{R}_{\geq 0}$ *such that*
  1. $\forall t \in T, \exists p \in P, \mathbf{W}(p,t) > 0$
  2. $\forall t \in T, \forall p \in P, \mathbf{C}(p,t) < 0 \Rightarrow \mathbf{W}(p,t) > 0$

A TDPN with an initial marking $\langle \mathcal{D}, \mathbf{m_0} \rangle$ *will be denoted a TDPN system.*

The first requirement about $\mathbf{W}$ ensures that the firing rate of any transition is defined, whereas the second one ensures that the marking remains non negative. Notice that these weights are defined as real numbers, while in TCPN all the arc weights are natural numbers.

We are now in position to give semantics to TDPNs.

Definition 5. *Let* $\mathcal{D}$ *be a TDPN. A trajectory is a continuously differentiable mapping* $\mathbf{m}$ *from time (i.e.,* $\mathbb{R}_{\geq 0}$*) to the set of markings (i.e.,* $(\mathbb{R}_{\geq 0})^P$*) which satisfies the following differential equation system:*

$$\begin{aligned} &\dot{\mathbf{m}} = \mathbf{C} \cdot \mathbf{f}(\mathbf{m}) \\ &\mathbf{f}(\mathbf{m})[t] = \min(\mathbf{W}(p,t) \cdot \mathbf{m}[p] \mid \mathbf{W}(p,t) > 0) \end{aligned} \tag{13}$$

Graphical notations. We extend the graphical notations of PN to take into account $\mathbf{W}$. These arcs are not oriented, since they are always defined as preconditions of transitions the orientation is implicit. Like in Forrester diagrams, to help distinguishing the $\mathbf{W}$ arcs from the **Pre** and **Post** arcs, $\mathbf{W}$ arcs will

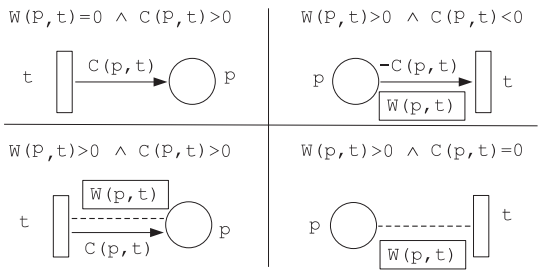

Fig. 6. Graphical notations: it will be seen that only the two on the top are needed

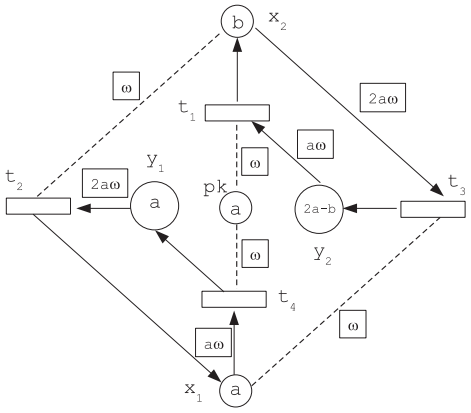

Fig. 7. A periodic TDPN

be drawn with dotted lines. To distinguish between the labels, $\mathbf{W}(p,t)$ will be drawn inside a box. There are four possible patterns illustrated in Fig. 6. When $\mathbf{W}(p,t) = 0 \wedge \mathbf{C}(p,t) > 0$, place p receives tokens from t and does not control its firing rate. When $\mathbf{W}(p,t) > 0 \wedge \mathbf{C}(p,t) < 0$, place p provides tokens to t. So it *must control* its firing rate. Hence, the non oriented arc between p and t is redundant, and we will not draw it and represent only an oriented arc from p to t both labelled by $-\mathbf{C}(p,t)$ and $\mathbf{W}(p,t)$. When $\mathbf{W}(p,t) > 0 \wedge \mathbf{C}(p,t) > 0$, place p receives tokens from t and controls its firing rate. There is both an oriented arc from t to p and a non oriented arc between p and t with their corresponding labels. When $\mathbf{W}(p,t) > 0 \wedge \mathbf{C}(p,t) = 0$, place p controls the firing rate of t and t does not modify the marking of p, so there is a non oriented arc between p and t. As usual, we omit labels when equal to 1.

As an example, the equations in (1) correspond to the TDPN in Fig. 7.

The Petri net structure of TCPN is similar to that of TDPN. Moreover, the minimum operator appears related to the marking evolution. Hence, it would not be surprising that both models are equivalent, as long as the elements of

the speed control matrix are in the rationals. To prove that, observe first that fractions in the **Pre** and **Post** matrices do not pose a problem, since they can be easily avoided multiplying the columns of the **Pre** and **Post** matrices by the right number (this does not change the dynamics of the system).

Proposition 1. *For any TCPN, a TDPN with the same number of places and transitions exists which has the timed evolution, and vice versa (as long as the weights of the TDPN are rational numbers).*

Proof. Let $\mathcal{D} = \langle P, T, \mathbf{C}, \mathbf{W} \rangle$ be a TDPN, and let us construct a TCPN $\mathcal{S} = \langle P, T, \mathbf{Pre}', \mathbf{Post}', \boldsymbol{\lambda} \rangle$ with the same differential equations. For each $t \in T$, define $\mathbf{W}^*(t) \geq \max(-\mathbf{C}(p,t) \cdot \mathbf{W}(p,t) \mid \mathbf{C}(p,t) < 0)$ (i.e., $\mathbf{W}^*(t)$ can be defined as any value that fulfills this inequality). By definition it is greater than zero. For each $p \in P$ and $t \in T$, define

- $\mathbf{Pre}'(p,t) = \mathbf{W}^*(t)/\mathbf{W}(p,t)$ if $\mathbf{W}(p,t) \neq 0$, and $\mathbf{Pre}'(p,t) = 0$ otherwise.
- $\mathbf{Post}'(p,t) = \mathbf{C}(p,t) + \mathbf{W}^*(t)/\mathbf{W}(p,t)$ if $\mathbf{W}(p,t) \neq 0$, and $\mathbf{Post}'(p,t) = \mathbf{C}(p,t)$ otherwise.
- $\boldsymbol{\lambda}(t) = \mathbf{W}^*(t)$

Let now $\mathcal{S} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{\lambda} \rangle$ be a TCPN, and let us construct a TDPN $\mathcal{D} = \langle P, T, \mathbf{C}', \mathbf{W} \rangle$ with the same differential equations. For each $p \in P$ and $t \in T$, define

- $\mathbf{C}'(p,t) = \mathbf{Post}(p,t) - \mathbf{Pre}(p,t)$
- $\mathbf{W}(p,t) = \boldsymbol{\lambda}(t)/\mathbf{Pre}(p,t)$ if $p \in {}^\bullet t$, and 0 otherwise. □

As an example, the TCPN in Fig. 4 and the TDPN in Fig. 7 are obtained one from the other with the previous procedure.

The patterns appearing at the bottom of Fig. 6 represent situations in which a place acts as a control place of a transition for which it is not an input place (in the sense that the transition is not removing tokens from the place). That is, they represent non-consuming control arcs. However, these patterns can be simulated with the two ones on top.

Proposition 2. *Let $\mathcal{D} = \langle P, T, \mathbf{C}, \mathbf{W} \rangle$ be a TDPN. Then another TDPN $\mathcal{D}' = \langle P', T', \mathbf{C}', \mathbf{W}' \rangle$ can be defined, using only the two patterns in the top of Fig. 6, with the same timed evolution as $\mathcal{D}$ but for some duplicated places. Moreover, the transformation is linear in size, since it at most doubles the number of places and transitions.*

Proof. Define the new TDPN as follows:

- $P' = \{p^-, p^+ \mid p \in P\}$
- $T' = \{t^-, t^+ \mid t \in T\}$
- To define $\mathbf{C}'$, distinguish two cases:
  - If $\mathbf{C}(p,t) < 0 \vee \mathbf{W}(p,t) = 0$ then
    $\mathbf{C}'(p^-, t^-) = \mathbf{C}'(p^+, t^+) = \mathbf{C}(p,t)$ and $\mathbf{C}'(p^-, t^+) = \mathbf{C}'(p^+, t^-) = 0$

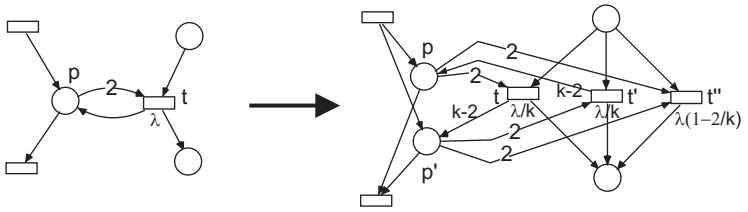

Fig. 8. TCPN can always be transformed into pure nets.

- If $\mathbf{C}(p,t) \geq 0 \wedge \mathbf{W}(p,t) > 0$ then
  $\mathbf{C}'(p^-,t^-) = \mathbf{C}'(p^+,t^+) = -1$ and $\mathbf{C}'(p^-,t^+) = \mathbf{C}'(p^+,t^-) = \mathbf{C}(p,t)+1$

– $\mathbf{W}'(p^-,t^-) = \mathbf{W}'(p^+,t^+) = \mathbf{W}(p,t)$ and $\mathbf{W}'(p^-,t^+) = \mathbf{W}'(p^+,t^-) = 0$ □

The transformation that has been proposed doubles the number of places and transitions. In practice, sometimes it is not necessary to duplicate all of them. A similar transformation can also be applied to *remove self-loop arcs* in TCPN. Let us see the idea in the case of one self-loop (see Fig. 8). The transformation replaces the self-loop with two places and three transitions. These two places will have the same marking the original place had, and the flow of the original transition is now split into the flow of these three transitions. Any other input/output place of the transition is input/output of all the transitions. Any input/output transition of the place is now input/output of all the places. The rates of these new transitions are defined so that the sum of their flows is equal to the flow of the original transition.

Again, this is a linear transformation. The procedure can be easily generalized to the case in which several places are connected with self-loops to the same transition (just duplicate each self-loop place and split the transition in three), or when one place is engaged in several self-loops (duplicate the self-loop place and split each transition in three).

Proposition 3. *For any TCPN a pure TCPN can be defined which has the same timed evolution, but for some duplicated places. Moreover, the transformation is linear in size, since it at most doubles the number of places and triples the number of transitions.*

The results have been summarized in the schema in Fig. 9.

Theorem 5. *The expressive power of TDPN, TDPN constrained to the use of the two basic constructions (the two on the top in Fig. 6), TCPN and pure TCPN are identical. Moreover, the transformations range from keeping the places and transitions to a linear increase in size (see the diagram in Fig. 9).*

5 Decidability Issues on Timed Continuous Petri Nets

In [6], it has been proved that TDPN can simulate two (non negative integer) counter machines (equivalent to Turing machines [7]). More precisely, different

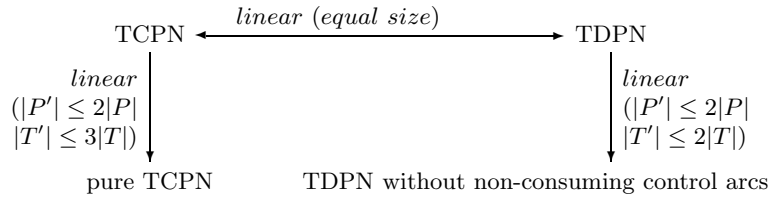

Fig. 9. Relationships between TDPN and TCPN and their versions without self-loops or non-consuming control arcs

simulations of two counter machines have been defined, in order to fulfill opposite requirements like *robustness* (allowing some perturbation of the simulation) and *boundedness* of the simulating net system (the simulation of an infinite-state system cannot be *simultaneously robust and bounded*). Furthermore these simulations can be performed by a net with a constant number of places, i.e., the dimension of the associated ordinary differential equation (ODE) is constant.

Theorem 6. *[6] Given a two counter machine $\mathcal{M}$, one can build a TDPN $\mathcal{D}$, with a constant number of places, whose size is linear w.r.t. the machine, with one of these two properties:*

- *its associated ODE has dimension* 6 *and $\mathcal{D}$ robustly simulates $\mathcal{M}$.*
- *it is a bounded net system, its associated ODE has dimension* 14*, and $\mathcal{D}$ simulates $\mathcal{M}$.*

Applying the equivalences developed in the previous section, the same result can be stated for TCPN with respect to the robust simulation. Regarding the bounded one, the proof in [6] requires the use of non rational numbers, and we have not been able to extend this result to TCPN up to now. As in [6], these results can be used to obtain undecidability results.

Corollary 3. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ be a TCPN whose associated ODE has dimension at least* 6*, $\mathbf{m}_1$ a marking, p a place and $k \in \mathbb{N}$. The problem whether there is a τ such that the trajectory starting at $\mathbf{m_0}$ fulfills*

- *$\mathbf{m}(\tau)(p) = k$ is undecidable.*
- *$\mathbf{m}(\tau)(p) \geq k$ is undecidable.*
- *$\mathbf{m}(\tau) \geq \mathbf{m}_1$ is undecidable.*

Corollary 4. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ be a TCPN whose associated ODE has dimension at least* 8*. Then the problem whether the trajectory $\mathbf{m}$ starting at $\mathbf{m_0}$ is such that $\lim_{\tau \to \infty} \mathbf{m}(\tau)$ exists, is undecidable.*

6 Conclusions

Regarding autonomous (i.e., untimed) nets, it has been proved that reachability, deadlock-freeness and liveness are decidable in continuous nets, both if

firing sequences are limited to be finite or if infinitely long sequences are allowed (Corollary 1, Theorem 3 and Theorem 4). Therefore, continuous and discrete autonomous nets are similar in decidability terms.

For timed nets, it has been proved that continuous pure nets are able to model Turing machines (Theorem 6). This means that many properties are undecidable (Corollary 3, Corollary 4). Timed continuous Petri nets can seen as a fluidified version of discrete Petri nets in which time is associated to transitions using a probability distribution function with positive mean value. Therefore, the fluidification has not increased decidability in the timed model.

However, fluidification *does* simplify some problems, and in particular some inverse problems. For example, the computation of an initial marking that maximizes a linear function of the throughput, the steady-state marking and the initial marking is easier for some nets in the continuous (timed) model than in the discrete one. More work is needed to characterize classes of problems that fluidification makes simpler, and why this happens.

References

1. H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8(1):159–188, 1998.
2. F. Balduzzi, A. Di Febraro, A. Giua, and C. Seatzu. Decidability results in first-order hybrid Petri nets. *Discrete Event Dynamic Systems*, 11(1–2):41–57, 2001.
3. G. Ciardo. Toward a definition of modeling power for stochastic Petri net models. In *Procs. of the Int. Workshop on PNPM*. IEEE-Computer Society Press, 1987.
4. J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *Bulletin of the EATCS*, 52:245–262, 1994.
5. Jay W. Forrester. *Principles of Systems*. Productivity Press, 1968.
6. S. Haddad, L. Recalde, and M. Silva. On the computational power of Timed Differentiable Petri nets. In E. Asarin and P. Bouyer, editors, *Formal Modeling and Analysis of Timed Systems, 4th Int. Conf. FORMATS 2006*, volume 4202 of *LNCS*, pages 230–244, Paris, 2006. Springer.
7. J.E. Hopcroft and J.D. Ullman. *Formal languages and their relation to automata*. Addison-Wesley, 1969.
8. J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In W. van der Aalst and E. Best, editors, *Proc. of the* 24^{th} *Int. Conf. on Application and Theory of Petri Nets (ICATPN 2003)*, volume 2679, pages 221–240. Springer, Eindhoven, The Netherlands, June 2003.
9. L. Recalde, E. Teruel, and M. Silva. Autonomous continuous P/T systems. In S. Donatelli and J. Kleijn, editors, *Application and Theory of Petri Nets 1999*, volume 1639 of *LNCS*, pages 107–126. Springer, 1999.
10. M. Silva. Introducing Petri nets. In *Practice of Petri Nets in Manufacturing*, pages 1–62. Chapman & Hall, 1993.
11. M. Silva and L. Recalde. Continuization of timed Petri nets: From performance evaluation to observation and control. In G. Ciardo and P. Darondeau, editors, *Procs. of ICATPN 2005*, volume 3536 of *LNCS*, pages 26–46. Springer, 2005.
12. M. Silva, E. Teruel, and J. M. Colom. Linear algebraic and linear programming techniques for the analysis of net systems. In G. Rozenberg and W. Reisig, editors, *Lectures in Petri Nets. I: Basic Models*, volume 1491 of *LNCS*, pages 309–373. Springer, 1998.