Annual Reviews in Control 28 (2004) 253–266

Annual Reviews in Control

www.elsevier.com/locate/arcontrol

On fluidification of Petri Nets: from discrete to hybrid and continuous models

Manuel Silva*, Laura Recalde[1]

Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, María de Luna 1, E-50018 Zaragoza, Spain

Received 9 March 2004; received in revised form 19 May 2004; accepted 20 May 2004

Abstract

Petri Nets (PNs) is a well-known modelling paradigm for discrete event systems (DES). As in other paradigms, hybrid and continuous PN formalisms have appeared in the literature, some of them being used in different engineering application domains. Hybridization may be obtained for example "by direct addition" of capabilities to model continuous subsystems. The approach adopted in this work is different: hybrid and continuous models appear because natural variables of a PN–DES model are transformed into non-negative reals. This relaxation may be quite reasonable when very populated or high traffic systems are considered. It is a classical relaxation applied to fight against the *state explosion problem* appearing when dealing with the analysis and synthesis of models. In tune with this, the paper presents "a biased" view of works in the hybrid PN arena. Partly an overview, this work revisits hybrid and continuous PNs, all of them being in essence hybrid models. Limitations to (partial) fluidification, and analysis and synthesis problems in this evolving field are considered. Several optimization problems (at design and at control) are also introduced here.

Keywords: Petri Nets; Relaxation; Hybrid models; Discrete event systems

1. Introduction

The Petri Nets *paradigm* is helpful for the modelling, analysis and synthesis of models, being useful at different phases of the life cycle of a system (see, e.g. Silva & Teruel, 1996). Nevertheless, like other different formal alternatives, enumeration techniques suffer from the so called *state explosion problem*, inherent to a large part of discrete event modelled systems. One way to deal with that problem is to use some kind of *relaxation*. *Fluidification* (or *continuization*) is a classical relaxation technique, in particular applicable to some discrete event models. In this setting, fluid models have potential for the application of more analytical techniques, possibly at the price of losing some modelling or analysis capabilities with respect to the discrete view.

Let us point out that the approach developed in this work follows the one introduced by R. David and H. Alla in Grenoble (Alla & David, 1998a; David & Alla, 1987), but there exist alternative hybrid PN formalisms around the idea of fluidification. Trivedi and his group introduced the so called *Fluid Petri Nets* (Horton, Kulkarni, Nicol, & Trivedi, 1998; Trivedi & Kulkarni, 1993). However, from a technical point of view the models are quite different. First let us point out that Trivedi and coworkers deal with hybrid models, not with simply fluid ones: only the marking of one (or a few) place is relaxed into the non-negative reals. Moreover, and contrary to the approach presented here, autonomous (untimed) models are not considered. Finally, timing remains for them stochastic, while in a first approximation we just take into account first moments (average values), fitting deterministic timing for the relaxed transitions.

Alternative approaches to hybrid PN models are: "direct addition" of continuous modelling capabilities, as done by Valette and coworkers (Champagnat, Esteban, Pingaud, & Valette, 1998), or some kind of reinterpretation as in

* Corresponding author.
E-mail address: silva@posta.unizar.es (M. Silva).
[1] Partially supported by projects FEDER and CICYT TIC2001-1819, and DPI2003-06376.

1367-5788/$ – see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.arcontrol.2004.05.002

Demongodin and Koussoulas (1998). For example, in the Valette approach systems of differential equations are associated to places. The set of marked places defines the continuous dynamic systems being active at a certain moment. Predicates at transitions may depend on continuous variables. Clearly, hybrid models may exhibit much more complex behaviours than discrete event or completely fluidified (i.e., "continuous") ones.

In our opinion, deep understanding of the kind of hybrid models obtained by partial fluidification of Petri Nets requires a better understanding of fluid or continuized models. Surprisingly enough, analysis of fully fluidified models does not receive still enough attention in the literature.

The existence of *non-linearizable* continuous models is a well known fact. Analogously, many Discrete Event Systems (DES) do *not* allow even partial continuization (e.g., due to the fact that qualitative properties as deadlock-freeness in the discrete model are neither necessary nor sufficient for deadlock-freeness of the continuous or hybrid-relaxed model; Recalde, Teruel, & Silva, 1999). Moreover, it happens that continuous timed models obtained from Petri Nets are *in essence hybrids*, more precisely *piecewise linear* in the basic cases of *infinite* and *single servers semantics* (Silva & Recalde, 2002). Under infinite server semantics, behavioural properties of continuous models that are not true for hybrid models are, for example, those preserved under marking scaling (*homothetic* behaviour). Under finite server semantics, marking scaling for the clients (i.e., excluding servers) has no effect on the performance of the model (*insensitivity* for clients). None of both is true, in general, for hybrid systems.

The present work is structured as follows: In Section 2 some basic concepts, and the notation to be used, are introduced. Continuous and hybrid nets are defined in Section 3. However, continuization does not always make sense, as can be seen in Section 4. The analysis of continuous and hybrid nets, both as autonomous and timed systems, is addressed in Section 5. Finally, the design and control of continuous timed models is considered in Section 6.

2. Basic concepts and notation

We assume that the reader is familiar with discrete PNs. For notation we use the standard one, see for instance Silva (1993).

A P/T system is a pair $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, where $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ is a P/T net (P and T are disjoint (finite) sets of *places* and *transitions*, and **Pre** and **Post** are $|P| \times |T|$ sized, natural valued, *incidence matrices*), and $\mathbf{m}_0$ is the *initial marking*.

For $v \in P \cup T$, the set of its input and output nodes are denoted as ${}^\bullet v$, and $v^\bullet$, respectively. A transition t is *enabled* at $\mathbf{m}$ iff for every $p \in {}^\bullet t$, $\mathbf{m}[p] > \mathbf{Pre}[p, t]$. Its *enabling degree* measures the maximal firing of the transition that can be done in one step, $\text{enab}(t, \mathbf{m}) = \lfloor \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p, t]\} \rfloor$. The firing of t in a certain amount $\alpha \in \mathbb{N}, \alpha \leq \text{enab}(t, \mathbf{m})$ 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*. If $\mathbf{m}$ is reachable from $\mathbf{m}_0$ through a sequence σ, a *state* (*or fundamental*) *equation* can be written: $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$, where $\sigma \in \mathbb{N}^{|T|}$ is the firing count vector.

Annullers of the incidence matrix are important because they induce certain invariant relations which are useful for reasoning on the behaviour. *Flows* (*semiflows*) are integer (natural) annullers of **C**. Right and left annullers are called T- and P-(semi)flows, respectively. For instance, if $\mathbf{y} \geq 0$ is such that $\mathbf{y} \cdot \mathbf{C} = 0$ then, every reachable marking $\mathbf{m}$ satisfies: $\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m}_0$. This provides a "token balance law". Analogously, if $\mathbf{x} \geq 0$ is such that $\mathbf{C} \cdot \mathbf{x} = 0$, then $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \mathbf{x} = \mathbf{m}_0$. That is, T-semiflows correspond to *potential* repetitive sequences. When $\mathbf{C} \cdot \mathbf{x} = 0$, $\mathbf{x} > 0$, the net is said to be *consistent*, and when $\mathbf{y} \cdot \mathbf{C} = 0, \mathbf{y} > 0$, the net is said to be *conservative*.

Two transitions, t and t', are in *structural conflict relation* iff ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$. The *coupled conflict relation* is defined as the transitive closure of the structural conflict relation. The set of all the induced equivalence classes is denoted by SCCS. Two transitions, t and t', are in *equal conflict* (EQ) relation when $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t'] \neq 0$. This is an equivalence relation and the set of all the equal conflict sets is denoted by SEQS. *Equal conflict systems* are systems in which all conflicts are equal, i.e., ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \text{Pre[P,t]=Pre[P,t']}$.

A P/T system is *bounded* when every place is bounded, i.e., its token content is less than some bound at every reachable marking. It is *live* when every transition is live, i.e., it can ultimately occur from every reachable marking. A net $\mathcal{N}$ is *structurally bounded* when $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ is bounded for *every* $\mathbf{m}_0$, and it is *structurally live* when *there exists* an $\mathbf{m}_0$ such that $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ is live.

Different timed extensions of (discrete) PNs have been introduced. Here we will assume that time is associated to the firing of transitions, and that either a deterministic or stochastic pdf is used. Vector $\boldsymbol{\lambda}$ defines the (mean) firing speeds. For the modelling of conflicts we may use *immediate* transitions with the addition of (marking and time independent) routing rates **R**. In other words, for the subset of immediate transitions $\{t_1, \ldots, t_k\} \subset T$ being in conflict at a reachable marking, we suppose that the constants $r_1, \ldots, r_k \in \mathbb{R}^+$ are explicitly defined in the system interpretation in such a way that when $t_1, \ldots, t_k$ are simultaneously enabled, transition t_i $(i = 1, \ldots, k)$ fires with probability $r_i / (\sum_{j=1}^{k} r_j)$. For a thorough discussion on the different policies used to solve conflicts among timed transitions see for example Ajmone Marsan et al. (1989).

3. Continuization as a standard relaxation

Simplifying the presentation for the actual purpose, it can be said that the analysis of Petri Nets models can be approached by *state enumeration* (e.g., reachability analysis and computation of the underlying Markov Chain), by *net transformation* (moving from a model to another one which is easier to analyse, while preserving the properties under study) or by *structural techniques* (using graph theory and/or mathematical programming techniques). Unfortunately transformation techniques are *not complete* (i.e., for every kit of transformation rules non-reducible net models exist), while structural analysis techniques, in general, only allow to *semi-decide* (either necessary or sufficient conditions are obtained), although for some net subclasses they allow to decide. Therefore, in some cases, the enumeration approach has to be used, with more or less sophisticated techniques (stubborn sets (Valmari, 1998), sleep methods, ...), probably leading to a state explosion problem.

Continuization (or fluidification) is one of the classical relaxations. The idea is analogous to that allowing the transformation of an integer linear programming problem (ILP, NP-hard) into a linear programming problem (LPP, polynomial complexity). This relaxation can be applied to Petri Nets in order to deal with the so called *state explosion problem*. The computational gain is usually increased if dealing with highly populated systems, because in those cases the state explosion problem may become much more acute. Fortunately, in most practical cases errors due to the relaxation of timed models happen to be not very significant when relatively heavy traffic conditions are relaxed.

Before advancing in this question, it should be pointed out that it has been proved by Recalde et al. (1999) and Silva and Recalde (2002) that full relaxation of an autonomous PN model, leading to so called *continuous* PNs, is essentially analogous to the use of convex geometry-linear programming techniques (as in Silva, Teruel, & Colom, 1998), which constitute a particular case of structural techniques.

Let us now consider timed formalisms, for example, Queuing Networks (QNs) which provide well known models for DES. Fluidification of QNs is a classical, and in many cases very practical, relaxation allowing not only analysis, but also synthesis of controls (see, e.g., Cassandras, Sun, Panayiotou, & Wardi, 2002; Chen & Yao, 2001;Mandelbau & Chen, 1991). In a quite different cultural setting, Forrester Diagrams (stock/flow diagrams), which appeared in the Systems Dynamics framework, is a well-known, essentially continuous, formalism for modelling certain classes of DES (Forrester, 1961, 1969). Comparisons of PNs and (mono-class) QNs or FDs can be seen in Vernon, Zahorjan, and Lazowska (1987), Silva and Campos (1993), Chiola (1998, chap. 4), and Jiménez, Recalde and Silva (2001), respectively. What is relevant here is that all three different formalisms are in essence *bipartite*:

	Reservoirs	Activities
PN	Places	Transitions
QN	Queues	Stations
FD	Deposits	Valves

From a structural point of view, the main differences of Petri Nets with respect to the other formalisms are the possible simultaneous existence in a single model of arc weights, attributions, choices, forks and joins, and the possible absence of local conservation rules (material, energy and information) when transitions are fired. Moreover, as it will be pointed out in Section 5.2, timed interpretation of the evolution may lead to different firing/flow policies. The firing logic of PNs is of the type *consumption/production*, a kind of generalization of the classical *client/servers* in QNs (Jackson, Gordon-Newell). Thus, continuization should be introduced through transitions, and extended to its neighbourhood (input and output places). When not all transitions are continuized, the obtained model is said to be *hybrid*. If all the transitions are continuized the net is said to be *"continuous"* (Alla & David, 1998a; Recalde & Silva 2001), even if, as it will be pointed out, time interpreted models exhibit a hybrid behaviour.

Observe that in the following definitions we do not consider *reading arcs* or *guards* (transitions just reading the state of a place; Montanari & Rossi, 1995), an extension of net models that escape from the consumption/production logic, but that sometimes are "inaccurately" modelled by self-loops (this means that tokens are consumed and immediately restored, not that a marking value is read). Probably the most interesting reading arcs, really extending the modelling power of (discrete) net models to that of Turing Machines (Agerwala & Flynn, 1973), are the so called *inhibitor arcs* or *zero test arcs*, in which the read variable acts under a negated logic (basically, if the place at origin is marked, the firing of the transition is inhibited). The inclusion of non-negated reading arcs is straightforward, but will complicate the notation, so it will not be considered here. The inclusion of inhibitor arcs presents some problems that go beyond the scope of this work. Thus, in order to simplify the presentation, just nets without guards and inhibitor arcs will be considered.

Definition 1. A *hybrid* (autonomous) Petri Net system is a pair $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, where

- $\mathcal{N}$ is a PN with two kinds of places and transitions, discrete and continuous, and such that places are discrete unless they are input or output of continuous transitions. That is, $\mathcal{N} = \langle P_D \cup P_C, T_D \cup T_C, \mathbf{Pre}, \mathbf{Post} \rangle$, with $P_C = {}^\bullet T_C \cup T_C^\bullet$, *and* **Pre** *and* **Post** $|P_D \cup P_C| \times |T_D \cup T_C|$ sized, natural valued, incidence matrices.
- $\mathbf{m}_0$ is a vector of markings of size $P_C \cup P_D$, and such that $\mathbf{m}_0[p] \in \mathbb{R}^+$ if $p \in P_C$ and $\mathbf{m}_0[p] \in \mathbb{N}$ *if* $p \in P_D$.

If $P_D = T_D = \emptyset$, it is a *continuous* Petri Net system. If $P_C = T_C = \emptyset$, it is a classical (discrete) Petri Net system.

As in discrete systems, the fundamental equation ($\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$, $\sigma \geq 0$) summarizes the way the marking evolves along time. But, in the continuous part, the marking is continuously changing, so we may consider the derivative of $\mathbf{m}$ with respect to time. If all the transitions are continuous ($T_D = \emptyset$) we obtain that $\dot{\mathbf{m}} = \mathbf{C} \cdot \dot{\boldsymbol{\sigma}}$. Let us call $\mathbf{f} = \dot{\boldsymbol{\sigma}}$, since it represents the *flow* through the transitions. Observe that it can be interpreted as the ''throughput vector'' of continuous PNs.

For the timing interpretation of continuous and hybrid PNs we will use a first order (or deterministic) approximation of the discrete case (Recalde & Silva, 2001), assuming that the delays associated to the firing of transitions can be approximated by their mean values. A similar approach is used, for example, in Balduzzi, Giua, and Menga (2000).

Different semantics have been defined for continuous transitions, the two most important being *infinite servers* (or *variable speed*) and *finite servers* (or *constant speed*) (Alla & David, 1998a; Recalde & Silva, 2001).

Under finite servers semantics, the flow of t, $\mathbf{f}[t]$ is defined as:

$$\mathbf{f}[t] = \begin{cases} \boldsymbol{\lambda}[t], & \text{if } \nexists\, p \in {}^{\bullet}t \quad \text{with } \mathbf{m}[p] = 0 \\ \min\left\{ \min_{p \in {}^{\bullet}t | \mathbf{m}[p]=0} \left\{ \sum_{t' \in {}^{\bullet}p} \dfrac{\mathbf{f}[t'] \cdot \mathbf{Post}[t', p]}{\mathbf{Pre}[p, t]} \right\}, \boldsymbol{\lambda}[t] \right\} \\ \text{otherwise} \end{cases}$$

Observe that then the flow of t is defined just using an upper bound, $\boldsymbol{\lambda}[t]$ (the number of servers times the speed of a server), knowing that at least one transition will be in saturation, that is, its utilization will be equal to 1.

Under infinite servers semantics, the flow through a timed transition t is the product of the speed, $\boldsymbol{\lambda}[t]$, and the enabling of the transition:

$$\mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \mathrm{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^{\bullet}t} \left\{ \frac{\mathbf{m}[p]}{\mathbf{Pre}[p, t]} \right\}$$

As in discrete PNs, immediate transitions act as flow splitters. The system obtained adding this equation to the state equation is positive (Farina & Rinaldi, 2000; Luenberger, 1979; Silva & Recalde, 2003), that is, $\mathbf{m} \geq 0$ is redundant.

In the association of a time semantics to the fluidification of a transition, it has been taken into account that a transition is like a station in QNs, thus ''the meeting point'' of *clients* and *servers*. Assuming that there may be many or few of each one of them, fluidification can be considered for clients, for servers or for both. Table 1 represents the four theoretically possible cases. Two of them do not allow the continuization, while the other two cases correspond to the previously introduced finite and infinite servers semantics.

Table 1
The four cases for possible continuization of a transition

Clients	Servers	Semantics of the transition
Few (D)	Few (D)	Discrete transition
Many (C)	Few (D)	Finite server semantics (bounds to firing speed)
Few (D)	Many (C)	Discrete transition (servers become *implicit places*)
Many (C)	Many (C)	Infinite servers semantics (speed is enabling-driven)

It should be pointed out that finite server semantics, equationally modelled by bounding the firing speed of continuized transitions, corresponds at pure conceptual level to a *hybrid* behaviour: fluidification is applied only to clients, while servers are kept as discrete, counted as a finite number (the firing speed is bounded by the product of the speed of a server and the number of servers in the station). More precisely, clients are assumed to be so many that they will never constraint the firing of transitions, although the net structure will impose some restrictions on the relative firing of the transitions (see *visit ratio* later on).

On the other hand, infinite servers semantics really relaxes clients and servers, being the firing speed driven by the enabling degree of the transition. In this case, although the fluidification is total, the model is hybrid in the sense that it is a piecewise linear system, in which switching among the embedded linear systems is not externally driven as in Bemporad, Giua, and Seatzu (2002), but internally through the minimum operators.

So, piecewise linear behaviours are obtained, either under finite or infinite servers semantics. In both cases the firing speed of a transition is defined through linear inequalities. If continuized models are obtained from *decoloration* of colored PN models (Ajmone Marsan & Neri, 1997; Chiola & Franceschinis, 1991; Dutheilett, Franceschinis, & Hadded, 1998, chap. 7), a product of state variables appears (as for the classical Volterra–Lotka model for a basic population dynamic case) (Silva & Recalde, 2002). This opens the door to the possibility of modelling *chaotic* behaviours with continuous PNs. In the following, unless otherwise stated, we will concentrate on basic, i.e., non-decolored, infinite servers semantics.

The system in Fig. 1 models a simple manufacturing system. A product is composed of two different parts, A and B, that are processed in machines $M1$ and $M2$, and stored in buffers BA and BB, respectively. Then, they are assembled by $M3$, and processed in $M4$. Finally, $M5$ packages them in twos. During all the processing of parts A and B, *Tool*1 and *Tool*2 are needed. Also *Tool*3 has to be used in the three final operations.

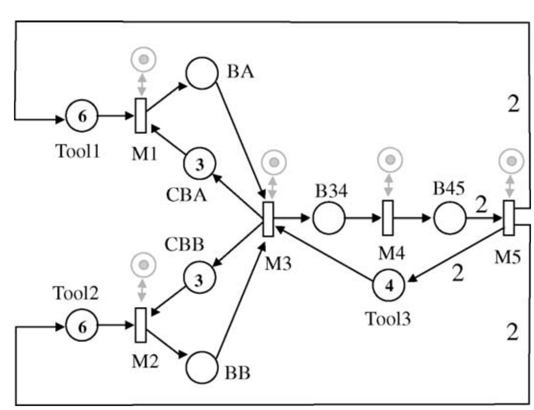

Fig. 1. A simple manufacturing system with firing rates [1, 1, 1, 1, 1, 1, 1].

Table 2
Throughput of $M5$ in the system represented at Fig. 1 when the initial marking is $k \cdot \mathbf{m}_0$, and $\mathbf{m}_0[Tool1] = \mathbf{m}_0[Tool2] = 6$, $\mathbf{m}_0[Tool3] = 4$, $\mathbf{m}_0[\mathrm{CBA}] = \mathbf{m}_0[C\mathrm{BB}] = 3$, and the rest unmarked

k	Size of the reachability space	Infinite servers semantics				Finite-server semantics (single server)
		Stations are "delays"		Stations with unique server		
		χ_I	χ_I/k	χ_I	χ_I/k	χ_S
1	205	0.4982	0.4982	0.2976	0.2976	0.2976
2	1885	1.1254	0.5627	0.7303	0.3652	0.3862
3	7796	1.7719	0.5906	1.1877	0.3959	0.4214
4	22187	2.4269	0.6067	1.6565	0.4141	0.4399
5	50801	3.0868	0.6174	2.1320	0.4264	0.4513
...	...	...	...	...	...	...
∞	∞	∞	2/3	∞	1/2	1/2

Let us see for this example how the continuized model relates to the discrete one under both semantics. Let us assume first that finite servers semantics is used. Table 2 shows how the value that is obtained for the throughput of the discrete net system increases when the marking is scaled, and approaches in the limit to the continuous one. If infinite servers semantics is applied instead, it is the throughput of the discrete net system divided by the multiplying constant which approaches to the continuous value.

In discrete PNs, *finite* servers semantics can be "simulated" using *infinite* servers semantics and adding self-loops to the transitions marked with the number of servers. This can also be done for continuous nets, but the result is not always the same as for finite servers (or constant speed) semantics. In continuous nets, finite servers semantics means that it is assumed that resources are never a restriction, and the only thing that restricts the evolution of the system is the speed of the servers. However, if finite servers are simulated with infinite servers semantics both things are taken into account. Hence, in the continuous system, depending on which is the most restrictive element (*servers* or *clients*), this infinite servers semantics with servers restrictions may behave as a finite servers semantics (see Table 2), or as infinite servers semantics (see Table 3).

Observe also that even if the steady state throughput of the two semantics is the same, the transient behaviour will be different. Moreover, even if the steady state throughput under both semantics is the same, the markings will be different.

In the following, whenever infinite servers semantics is applied to this example, we will assume that the machines restrictions are included in the model, i.e., it is the model with the self-loops that we will be studying.

Visit ratios measure the relative flow or throughput of customers in QNs (with respect to a given station). In (discrete) PNs visit ratios generalize the notion in order to measure the relative (to a transition) throughput of tokens (Campos & Silva, 1992). Let $\boldsymbol{\chi}$ be throughput vector of the net system, that is, $\boldsymbol{\chi}[t_j]$ is the number of firings per time unit of transition t_j. If transition t_i is taken as a reference, the relative throughput of t_j is the so called *visit ratio* of t_j with respect to t_i: $\mathbf{v}^{(i)}[t_j] = \boldsymbol{\chi}[t_j]/\boldsymbol{\chi}[t_i]$. Notice that, due to the way it is defined, $\mathbf{v}^{(i)}[t_i] = 1$. In general, the vector of visit ratios, $\mathbf{v}$, is a function of the net structure, the initial marking, the routing (controlling conflict resolutions) and the service time of transitions: $\mathbf{v} = \mathbf{v}(\mathcal{N}, \mathbf{m}_0, \mathbf{R}, \boldsymbol{\lambda})$, both for discrete and continuous net systems. The computability of visit ratios induces somehow a hierarchy of nets, and some well-known net subclasses are "re-encountered" (see Table 4).

For example, mono-T-semiflow nets are those that have only one full minimal T-semiflow. That is, the system $\mathbf{C} \cdot \mathbf{x} = 0$, has only one positive solution, up to constants. This means that the vector of visit ratios is completely defined by the net structure, since it must fulfill that $\mathbf{C} \cdot \mathbf{v}^{(1)} = 0$ and $\mathbf{v}^{(1)}[t_1] = 1$. This happens for instance in the example of Fig. 1. For this net all the T-semiflows are proportional to $[2, 2, 2, 2, 1]$. And so, the visit ratio w.r.t. $M5$ is $[2, 2, 2, 2, 1]$, not depending on anything but the net structure.

Table 3
Throughput of $M5$ in the system represented at Fig. 1 when the initial marking is $k \cdot \mathbf{m}_0$, and $\mathbf{m}_0$ is as in the figure except that $\mathbf{m}_0[Tool3] = 2$

k	Size of the reachability space	Infinite servers semantics				Finite-server semantics (single server)
		Stations are "delays"		Stations with unique server		
		χ_I	χ_I/k	χ_I	χ_I/k	χ_S
1	96	0.2650	0.2650	0.2169	0.2169	0.2169
2	735	0.5852	0.2926	0.5285	0.2642	0.3364
3	2800	0.9166	0.3055	0.8592	0.2864	0.3846
4	7605	1.2498	0.3125	1.1946	0.2987	0.4140
5	16896	1.5832	0.3166	1.5317	0.3063	0.4318
...	...	...	...	...	...	...
∞	∞	∞	1/3	∞	1/3	1/2

Table 4
Computation of visit ratios in net subclasses, adapted from Campos and Silva (1992)

Net system subclass	Visit ratio, $\mathbf{v}$
Strongly connected marked graphs	$\mathbf{1}$, Constant
Mono-T-semiflow nets	$\mathbf{v}(\mathcal{N})$
Mono-T-semiflow reducible nets	$\mathbf{v}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{R})$
Simple nets	$\mathbf{v}(\mathcal{N}, \mathbf{m}_0, \mathbf{R}, \boldsymbol{\lambda})$

Mono-T-semiflow reducible nets is a generalization, defined as those PNs for which the following system has a unique solution:

$$\begin{array}{ll} \mathbf{C}\cdot\mathbf{v}^{(1)} = 0\,\mathbf{v}^{(1)} > 0 & \\ \mathbf{R}\cdot\mathbf{v}^{(1)} = 0, & \text{for the resolution of EQ conflicts among immediate transitions} \\ \dfrac{\mathbf{v}^{(1)}[t_i]}{\boldsymbol{\lambda}[t_i]} = \dfrac{\mathbf{v}^{(1)}[t_j]}{\boldsymbol{\lambda}[t_j]}, & \forall\, t_i, t_j \textit{ timed} \text{ transitions in EQ relation} \end{array} \quad (1)$$

where $\mathbf{R}$ is the routing rates matrix. That is, $\mathbf{R}\cdot\mathbf{v}^{(1)} = 0$ is equivalent to $(\mathbf{v}^{(1)}[t_i]/r_i) = (\mathbf{v}^{(1)}[t_j]/r_j)$ for every pair of immediate transitions in EQ relation.

For example, structurally live and bounded EQ nets with conflicts solved through immediate transitions belong to the class. Observe that in mono-T-semiflow reducible nets the visit ratio only depends on $\mathcal{N}$, $\mathbf{R}$ and $\boldsymbol{\lambda}$, i.e., the structure and interpretation, but not on $\mathbf{m}_0$.

4. Partial continuization is not always possible

If a discrete net model is to be partially continuized, a basic question appears: *which* are the transitions that is reasonable to fluidify? This question does not have now a complete answer at general level, and only guidelines, not rules, can be provided. Relaxation of a given model, like modelling in itself, is partially an art, and in many cases it requires ingenuity and experience, something essential in most engineering activities. For autonomous (untimed) net models, guidelines should take into account the token load of places (those having usually many tokens are better candidates). For timed models, the driving idea is that errors due to the relaxation tend to be not very significant under relatively heavy traffic conditions. Therefore, as a guideline, not as a rule, it can be said that transitions with relatively high visit ratio are "good" candidates for fluidification.

However, this does not work for all the systems. To begin with, not all Petri Net systems allow continuization as an approximate modelling, and important technical problems may sometimes appear (Recalde et al., 1999).

For example, the autonomous system in Fig. 2 is live as discrete. However, if it is seen as continuous or hybrid (perhaps t_1 and t_3 should be kept as discrete), it is non-live. In a similar way, a discrete bounded system may be unbounded as continuous or hybrid.

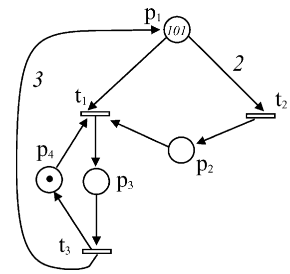

Fig. 2. A system that is live as discrete, but non-live as continuous or hybrid (with t_2 as continuous).

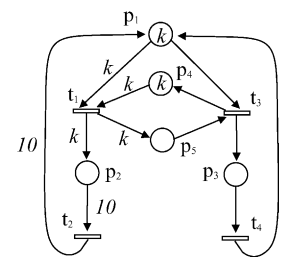

Fig. 3. The unique T-semiflow is $\mathbf{x} = [1, k/10, k, k]$, thus "a priori" assuming $k > 10$ the less reasonable candidate to fluidify is t_1.

Problems may also be related to which are the transitions that are continuized or the firing semantics that is being used. To simplify, here we will focus on timed transitions, although similar problems may appear if immediate transitions are allowed.

Consider the system in Fig. 3, with firing speed vector $\boldsymbol{\lambda} = [1, 3, 30, 30]$, under infinite servers semantics. For this net, $\mathbf{v}^{(1)} = [1, k/10, k, k]$ and, if $k \gg 10$, t_1 will be fired far less than the others. Moreover, taking into account that it is used for the synchronization of many tokens, it seems reasonable to leave it as discrete, and continuize only t_2, t_3 and t_4. However, if it is done so the throughput of the hybrid system tends to zero! This is due to the queue of the exponential decay. On the other hand, if all the transitions are continuized, the throughput is on average 0.732 (optimistic!), but quite close to the value of the discrete system both if the transitions are defined as deterministic (0.714) or have exponential pdf (0.602). However, if finite servers semantics is used, the hybrid system has average throughput 0.5, the continuous system 1, the discrete system with exponential transitions 0.677, and 0.423 with deterministic timing (see Table 5). The problem with the hybrid interpretation under infinite servers semantics is due to the fact that all the tokens have to be synchronized in order to

Table 5
Throughput differences in the system in Fig. 3

	Single server	Infinite servers
Discrete	0.423 (deterministic)	0.714 (deterministic)
	0.677 (Markovian)	0.602 (Markovian)
Continuous	1	0.732
Hybrid (t_1 discrete)	0.5	0

fire t_1, but it takes infinite time to put them in p_1 and p_4 (the problem is similar to the charge of a condenser in an R–C circuit). If the marking of p_1 and p_4 is increased by one token, the average throughput of the hybrid system changes from zero to 0.4686. In fact, adding a very small number of tokens to these places the throughput increases quickly.

5. On the structural analysis of hybrid and continuous net models

Structural analysis tries to get the maximal amount of information from the net structure, and sees $\mathbf{m}_0$ as a parameter. Here we will focus on mathematical programming techniques, and not in graph based ones (for discrete nets see Silva, Teruel, & Colom, 1998).

From the adopted perspective, at a purely conceptual level, there exists a kind of hierarchy from discrete to hybrid and continuous models. Usually, there is a progressive degradation from discrete to continuous, through hybrid, in the quality of the model. This is reflected in a parallel improvement in the cost of the "same" analysis techniques. But relaxation means also that some properties are lost. For example, mutex properties cannot be observed in continuous systems (and just partially in hybrid ones), and existence of home states is equivalent to reversibility. Other properties have to be modified, for example lim-properties (Recalde et al., 1999), or δ-properties (Júlvez, Recalde, & Silva, 2003).

5.1. Autonomous models

The fundamental equation ($\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$, $\sigma \geq 0$) can be used in the continuous (or hybrid) formalism, with the only difference that variables belong to $\mathbb{R}^+$ (some to $\mathbb{R}^+$ and some to $\mathbb{N}$, in the hybrid case). Moreover, there are no spurious solutions in continuous models, under very light conditions (Júlvez et al., 2003; Recalde et al., 1999), but they start to appear in hybrid models.

This means that the fundamental equation is extremely useful for the analysis of properties. For example, a simple necessary and sufficient condition for deadlock-freeness can be obtained for continuous PNs: a solution of the fundamental equation exists in which no transition is enabled.

Two characteristics of continuous models that do not hold for discrete models are that the set of reachable markings is a *convex set*, and that it verifies a *scaling property*: if $\mathbf{m}$ is reachable in $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $k \cdot \mathbf{m}$ is reachable in $\langle \mathcal{N}, k \cdot \mathbf{m}_0 \rangle$ (Recalde et al., 1999). This is less and less true for hybrid models, and certainly not true for discrete models. However, neither continuous, nor hybrid or discrete, are live-monotonic with respect to the marking. That is, increasing the initial marking (i.e., the number of resources) can kill the system (Recalde, Júlvez, & Silva, 2002).

P- and T-semiflows can be used in continuous or hybrid models just as in discrete ones. In discrete systems *rank theorems* provide necessary or sufficient structural conditions for liveness and boundedness, that are both necessary and sufficient for some subclasses. These results can be generalized to continuous or hybrid systems, since they are based on an underlying "continuous-view" of net systems.

Theorem 2 (Recalde et al., 1999; Teruel & Silva, 1996).

- *Let* $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ *be a live and bounded system. Then,* $\mathcal{N}$ *is consistent, conservative and* rank $(\mathbf{C}) \leq |\text{SEQS}| - 1$.
- *Let* $\mathcal{N}$ *be a consistent and conservative net. If* rank $(\mathbf{C}) = |\text{SCCS}| - 1$, then $\mathcal{N}$ *is structurally live.*

For EQ systems the rank theorem is a characterization of structural liveness and structural boundedness.

Theorem 3 (Recalde et al., 1999; Teruel & Silva, 1996). *An EQ net is structurally live and structurally bounded if and only if it is consistent, conservative and rank*$(\mathbf{C}) = |\text{SEQS}| - 1$.

Traps (a set of places whose output is contained in its input, i.e., $\Theta \subseteq P, \Theta^\bullet \subset {}^\bullet\Theta$), and *siphons* (a set of places whose input is contained in its output, i.e., $\Sigma \subseteq P, {}^\bullet\Sigma \subset \Sigma^\bullet$) can also be defined in the continuous or hybrid case, since they are structural objects. However, some of the behavioural properties they enjoy in discrete models are lost for continuous: traps can be emptied (at the limit), and "almost" empty siphons can recover, even if the system is bounded.

5.2. Timed models under infinite servers semantics

Under finite servers semantics, the *evolution graph* (Alla & David, 1998b) has been defined as a way to represent how the systems of equations that define the evolution of the system change. A similar graph could be defined for infinite servers semantics. This is basically a behavioural approach. Our approach here will be more structural.

In continuous PNs, as in discrete PNs, the throughput of the system is *non-monotonic* with respect to $\boldsymbol{\lambda}$ (i.e., increasing the speed of a transition may lead to a decrease in the throughput), neither w.r.t. $\mathbf{m}_0$ (i.e., increasing the number of resources may lead to a decrease in the throughput), nor w.r.t. the net structure (i.e., adding places, restrictions, may increase the throughput) (Recalde et al., 2002).

It could be thought that, since continuization removes some restrictions of the system, the throughput of the continuous system should be at least that of the discrete one. However, the throughput of a continuous PN is *not* in general optimistic, i.e., an upper bound of the throughput of the discrete PN. It can also be a pessimistic approximation (see Fig. 4).

For bounded *discrete* net systems, an upper bound of the throughput of one transition ($\boldsymbol{\phi}[t_i] \geq \boldsymbol{\chi}[t_i]$) can be obtained by means of the following linear programming problem (Campos, 1998; Chiola, Anglano, Campos, Colom, & Silva, 1993).

$$\begin{aligned} &\max\{\boldsymbol{\phi}[t_i] \\ &\text{s.t. } \boldsymbol{\mu} = \mathbf{m}_0 + \mathbf{C}\cdot\sigma \\ &\quad \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall p \in {}^\bullet t \\ &\quad \mathbf{R}\cdot\boldsymbol{\phi} = 0 \\ &\quad \mathbf{C}\cdot\boldsymbol{\phi} = 0, \\ &\quad \boldsymbol{\mu}, \sigma, \boldsymbol{\phi} \geq 0\} \end{aligned} \tag{2}$$

These equations correspond to the fundamental equation (a necessary condition for $\boldsymbol{\mu}$ to be the average marking), the Little Law applied to the different transitions, the routing at the free-conflicts, and the fact that $\boldsymbol{\phi}$ has to be a T-semiflow (can be replaced by $\mathbf{C}\cdot\boldsymbol{\phi} \geq 0$ if the net is not structurally bounded). The linear programming problem is exact (i.e., $\boldsymbol{\chi} = \boldsymbol{\phi}$) in some cases, for example, for deterministic marked graphs, state machines with just delays, ...

For *continuous* (and *hybrid*) models almost the same linear programming problem can be written. The only difference is that the flow law at continuous persistent transitions (with a single input place) can be improved. Let us denote as T_{U} the set of transitions with a unique input place, and as T_{S} the remaining transitions, in which synchronizations appear. Using the same kind of notation as before,

- $\boldsymbol{\phi}$ as a the approximation of $\boldsymbol{\chi}$,
- $\boldsymbol{\mu}$ as the approximation of $\mathbf{m}$

it can be stated as follows:

$$\begin{aligned} &\max\{\boldsymbol{\phi}[t_i] \\ &\text{s.t. } \boldsymbol{\mu} = \mathbf{m}_0 + \mathbf{C}\cdot\sigma \\ &\quad \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]}, \qquad \forall t \in T_{\mathrm{S}}, \forall p \in {}^\bullet t \\ &\quad \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]}, \qquad \forall t \in T_{\mathrm{U}}, p = {}^\bullet t \\ &\quad \mathbf{R}\cdot\boldsymbol{\phi} = 0 \\ &\quad \mathbf{C}\cdot\boldsymbol{\phi} = 0 \\ &\quad \boldsymbol{\mu}, \sigma, \boldsymbol{\phi} \geq 0\} \end{aligned} \tag{3}$$

In general the solution of (3) is just an upper bound, although it is exact for example for EQ nets (Recalde & Silva, 2001), or some mono-T-reducible nets (Recalde et al., 2002).

Observe that the LPP (3) is in fact a relaxation of the following *nonlinear* (due to the "*min*" operators) programming problem:

$$\begin{aligned} &\max\{\boldsymbol{\phi}[t_i] \\ &\text{s.t. } \boldsymbol{\mu} = \mathbf{m}_0 + \mathbf{C}\cdot\sigma \\ &\quad \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t]\cdot\min_{p \in {}^\bullet t}\left\{\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]}\right\}, \qquad \forall t \in T \\ &\quad \mathbf{R}\cdot\boldsymbol{\phi} = 0 \\ &\quad \mathbf{C}\cdot\boldsymbol{\phi} = 0 \\ &\quad \boldsymbol{\mu}, \sigma, \boldsymbol{\phi} \geq 0\} \end{aligned} \tag{4}$$

Unfortunately, programming problem (4) cannot be solved in general with a polynomial time algorithm. A branch and bound based algorithm can be used to solve it (Recalde et al., 2002). The idea is to first solve the relaxed LPP defined in (3). Then, if the marking does not correspond to a *steady state* (i.e., there is at least one transition such that all its input places have "too many" tokens) choose one of the synchronizations and solve the set of LPPs that appear when each one of the input places is assumed to be defining the flow. That is, build a set of LPPs by adding an equation that relates the marking of each input place with the flow of the transition. These subproblems become children of the root search node. The algorithm is applied recursively, generating a tree of subproblems. If an optimal steady state marking is found to a subproblem, it is a possible steady state marking, but not necessarily globally optimal. Since it is feasible, it can be used to prune the rest of the tree; if the solution of the LPP for a node is smaller than the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned.

For mono-T-semiflow reducible nets, problem (3) can be simplified. In this class, $\mathbf{v}^{(1)}$ is completely defined (see Eq. (1), and since $\boldsymbol{\chi}$ is proportional to $\mathbf{v}^{(1)}$, $\boldsymbol{\chi} = \alpha\cdot\mathbf{v}^{(1)}$. Hence, maximizing $\boldsymbol{\chi}[t]$ is equal to maximizing α. Moreover, the equalities added to LPP (3) with respect to LPP (2) can be removed, since if the obtained solution does not fulfill them, there exists another one with the same throughput which does. The idea is to send the extra marking "down" the transition till a synchronization appears. Since it is a mono-T-semiflow net, there cannot be cycles without synchronizations, unless the whole system is just a cycle, and in that case all the equalities must be fulfilled in the maximum. Applying all this to the LPP (3), $\boldsymbol{\phi} \geq \boldsymbol{\chi}$ can be obtained as follows:

$$\begin{aligned} &\boldsymbol{\phi}[t_i] = \max\{\alpha \\ &\quad \text{s.t. } \boldsymbol{\mu} = \mathbf{m}_0 + \mathbf{C}\cdot\sigma \\ &\quad\quad \alpha\cdot\mathbf{PD}^{(1)} \leq \boldsymbol{\mu} \\ &\quad\quad \boldsymbol{\mu}, \sigma \geq 0\} \end{aligned} \tag{5}$$

where $\mathbf{PD}^{(1)}$ is defined as

$$\mathbf{PD}^{(1)}[p] = \max_{t \in p^\bullet}\left\{\frac{\mathbf{Pre}[p,t]\cdot\mathbf{v}^{(1)}[t]}{\boldsymbol{\lambda}[t]}\right\}$$

Applying linear programming duality and changing variables, the following LPP can be written, which is clearly related to the "classical" one for discrete net systems in Campos, Chiola, and Silva (1991):

$$\begin{aligned} &\frac{1}{\boldsymbol{\phi}[t_i]} = \max\{\mathbf{y}\cdot\mathbf{PD}^{(1)} \\ &\quad \text{s.t. } \mathbf{y}\cdot\mathbf{C} = 0 \\ &\quad\quad \mathbf{y}\cdot\mathbf{m}_0 = 1 \\ &\quad\quad \mathbf{y} \geq 0\} \end{aligned} \tag{6}$$

For example, for the net system in Fig. 1 (with the self-loops around the machines), if $\mathbf{m}_0[Tool3] = 4$, the result of this LPP is 1/2, which comes from the use of the machines. If $\mathbf{m}_0[Tool3] = 2$, the result is 1/3, which derives from the P-semiflow associated to the use of Tool3.

6. On the synthesis of continuous timed net models: design and control problems

In order to limit our presentation here, let us assume that the system is already logically constrained by the required controller (*generalized mutual exclusion controller*, GMEC (Giua, DiCesare, & Silva, 1993), *supervisory controller* (Ramadge & Wonham, 1989), ...).

6.1. Design problems: the EQ systems case

In this section we consider "off-line" problems in which, given the system configuration, it is optimally parameterized for the steady state. Among the problems belonging to this class are those devoted to the computation of an *optimal* $\mathbf{m}_0$, i.e., problems in which the goal is the search of the optimal number of resources, like the best number of machines or AGVs, or the optimal size of buffers. Other kind of problems belonging to this general scheme are those in which the goal is the computation of the *optimal routing*, $\mathbf{R}$, or the *optimal firing speeds*, $\boldsymbol{\lambda}$. Examples of this last pattern are equipment selection problems (choice of best machines). Close to this framework, some steady state control problems are considered in Section 6.4.

6.2. Optimization of $\mathbf{m}_0$

A quite frequent formulation for this class of optimization problems in steady state is: try to *maximize a profit function* depending on the flow or throughput vector ($\boldsymbol{\chi}$), the average marking ($\mathbf{m}_{ss}$), and the initial marking ($\mathbf{m}_0$), among other variables. If the profit function may be formulated in linear terms, it may adopt patterns like: $\mathbf{g} \cdot \boldsymbol{\chi} - \mathbf{w} \cdot \mathbf{m}_{ss} - \mathbf{b} \cdot \mathbf{m}_0$, where, $\mathbf{g}$ represents a gain vector w.r.t. to flows (e.g., if $\boldsymbol{\chi}[t_1]$ is to be maximized, $\mathbf{g}[t_1] = 1$, while the rest of weights should be zero), $\mathbf{w}$ is the *cost* vector due to immobilization to maintain the production flow (e.g., due to the levels in stores), and vector $\mathbf{b}$ represent *depreciations or amortization of the initial investments* (e.g., due to the size of the stores, number of machines, ...). In other cases, the optimization tries to *minimize a cost function* (e.g. a weighting on the initial marking, $\mathbf{v} \cdot \mathbf{m}_0$).

This kind of optimization problems admit a particularly elegant and efficient solution if the LPPs, stated in Section 5.2 lead to the exact value (otherwise *upper bounds are obtained*). As was previously mentioned, this happens, for example, for structurally live and bounded EQ nets (its characterization can be computed polynomially through the rank theorem, see Section 5.1). Let us consider some interesting and simple optimization problems. For simplicity, in the sequel of this section *let us assume that nets are structurally live and bounded EQ* (*thus, mono-T-semiflow reducible*), *and conflicts among immediate transitions are solved according to routing rates,* $\mathbf{R}$.

Problem 4 (*Silva & Recalde, 2002*).
Given $\mathbf{V} \cdot \mathbf{m}_0 \leq \mathbf{k}$, as a set of linear cost-constraints on the initial marking, compute an optimal $\mathbf{m}_0$ in order to maximize $\boldsymbol{\phi}[t_i]$ (here $\mathbf{g} = \mathbf{1}_i, \mathbf{w} = \mathbf{0}$ and $\mathbf{b} = \mathbf{0}$).

Simply adding the cost constraints to LPP (5), the following can be written, in which $\boldsymbol{\mu}_0$ denotes the approximation of $\mathbf{m}_0$:

$$
\begin{aligned}
\boldsymbol{\Gamma}[t_i] = \max\{ & \alpha \\
\text{s.t.}\ & \alpha \cdot \mathbf{PD}^{(1)} \leq \boldsymbol{\mu}_0 + \mathbf{C} \cdot \sigma \\
& \mathbf{V} \cdot \boldsymbol{\mu}_0 \leq \mathbf{k} \\
& \boldsymbol{\mu}_0, \sigma \geq 0\}
\end{aligned}
\tag{7}
$$

If LPP (3) is used instead as the basis (it is equivalent if the visit ratio is computable with $\mathbf{C} \cdot \boldsymbol{\phi} = 0$ and $\mathbf{R} \cdot \boldsymbol{\phi} = 0$) (Campos, 1998, chap. 17), the optimization problem can be solved by means of the following alternative LPP, that is also straightforward to derive; only the cost constraints on the initial marking, now a vector of variables, are *added*. Nevertheless a higher number of (in)equations and variables appear:

$$
\begin{aligned}
\max\{ & \boldsymbol{\phi}[t_i] \\
\text{s.t.}\ & \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C} \cdot \sigma \\
& \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t \in T_{\mathrm{S}}, \forall p \in {}^{\bullet}t \\
& \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t \in T_{\mathrm{U}}, p = {}^{\bullet}t \\
& \mathbf{R} \cdot \boldsymbol{\phi} = 0 \\
& \mathbf{C} \cdot \boldsymbol{\phi} = 0 \\
& \boldsymbol{\sigma}, \boldsymbol{\mu}_0 \geq 0 \\
& \mathbf{V} \cdot \boldsymbol{\mu}_0 \leq \mathbf{k}\}
\end{aligned}
\tag{8}
$$

When the net system is not EQ, the LPPs in (7) and (8) provide just bounds (a lower bound of the cycle time in (7) and an upper bound of the throughput of a transition in (8)).

Problem 5 (*Silva & Recalde, 2002*).
Given a cost weight vector $\mathbf{b}$ and a cycle time $\Gamma_i = 1/\boldsymbol{\phi}[t_i]$, compute the minimal cost initial marking, $\mathbf{b} \cdot \boldsymbol{\mu}_0$, to have the given cycle time:

$$
\begin{aligned}
\min\{ & \mathbf{b} \cdot \boldsymbol{\mu}_0 \\
\text{s.t.}\ & \mathbf{PD}^{(1)} \leq \mathbf{C} \cdot \mathbf{z} + \Gamma_i \cdot \boldsymbol{\mu}_0 \\
& \mathbf{z}, \boldsymbol{\mu}_0 \geq 0\}
\end{aligned}
\tag{9}
$$

Let us apply this LPP to the system in Fig. 1, to see which is the best initial marking that allows to obtain a throughput of 1/2, i.e., cycle time 2. Assume that the costs vector is 10 per tool, i.e., $\mathbf{b}[Tool1] = \mathbf{b}[Tool2] = \mathbf{b}[Tool3] = 10$, 5 per buffer space, i.e., $\mathbf{b}[CBA] = \mathbf{b}[CBB] = 5$, and w.r.t. intermediate stocks $\mathbf{b}[\mathrm{BA}] = \mathbf{b}[\mathrm{BB}] = 20$, and $\mathbf{b}[B34] = \mathbf{b}[B45] = 40$. Then, the marking that is obtained is $\mu[Tool1] = \mu[Tool2] = 4$, $\mu[Tool3] = 3$ and $\mu[C\mathrm{BA}] = \mu[C\mathrm{BB}] = 2$.

Alternatively, if LPP (3) is used instead as the basis, the optimization problem can be solved by means of the following LPP, that once again is straightforward to derive; only the last constraint, on the lower bound for the flow of t_i, should be *added*.

$$\begin{aligned}
&\min\{\mathbf{b}\cdot\boldsymbol{\mu}_0\\
&\text{s.t. } \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C}\cdot\sigma\\
&\quad \boldsymbol{\phi}[t] \le \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t\in T_{\mathrm{S}},\ \forall p\in{}^{\bullet}t\\
&\quad \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t\in T_{\mathrm{U}}, p={}^{\bullet}t\\
&\quad \mathbf{R}\cdot\boldsymbol{\phi} = 0\\
&\quad \mathbf{C}\cdot\boldsymbol{\phi} = 0\\
&\quad \sigma, \boldsymbol{\mu}_0 \ge 0\\
&\quad \boldsymbol{\phi}[t_i] \ge 1/\Gamma_i\}
\end{aligned} \tag{10}$$

6.3. Optimization of routing, R

A simple case for optimizing a profit function w.r.t. the routing $\mathbf{R}$ is the following example. As in the preceding section, *let us assume that* $\langle\mathcal{N}, \mathbf{m}_0\rangle$ *is a structurally live and bounded EQ system.*

Problem 6. Maximize $\mathbf{g}\cdot\boldsymbol{\phi} - \mathbf{w}\cdot\boldsymbol{\mu} - \mathbf{b}\cdot\boldsymbol{\mu}_0$, with respect to the routing.

The following LPP computes an optimal flow vector, $\boldsymbol{\phi}$. It can be remarked that the only difference w.r.t. LPP (3) has been to *remove* the constraint on the routing.

$$\begin{aligned}
&\max\{\mathbf{g}\cdot\boldsymbol{\phi} - \mathbf{w}\cdot\boldsymbol{\mu} - \mathbf{b}\cdot\boldsymbol{\mu}_0\\
&\text{s.t. } \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C}\cdot\sigma\\
&\quad \boldsymbol{\phi}[t] \le \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t\in T_{\mathrm{S}},\ \forall p\in{}^{\bullet}t\\
&\quad \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \qquad \forall t\in T_{\mathrm{U}}, p={}^{\bullet}t\\
&\quad \mathbf{C}\cdot\boldsymbol{\phi} = 0\\
&\quad \sigma, \boldsymbol{\mu}_0 \ge 0\}
\end{aligned} \tag{11}$$

Once LPP (11) has been solved, the computation of the routing matrix $\mathbf{R}$ is straightforward: just proceed free-choice by free-choice. Assuming for simplicity that choices are binary:

- $\frac{\phi_1}{\phi_2} = \frac{r_1}{r_2}$
- $r_1 + r_2 = 1$

Note, if all conflicts are solved with immediate transitions, and $\mathbf{g} = \mathbf{1}, \mathbf{w} = \mathbf{b} = 0$, this LPP is analogous to the one stated in Gaujal and Giua (2002), assuming boundedness. Even if in this case nets are P-timed with a delay associated to places, and conflicts are solved according to a stationary routing policy (which in practice is equivalent a net without conflicts), and have different transient behaviour, their steady state is the same.

6.4. Control problems in general net systems

In this section the system configuration and parameters are supposed to be fixed. The problems to be considered here are those devoted to the computation of some *dynamic control variables*. Among many examples that can be presented, we deal with the computation of a *feed forward* control for maintaining an optimal steady state, or for bringing the system from an initial marking to another final marking in minimal time. Because the control of equal conflict net systems is relatively easy, let us consider in the sequel *general* continuous PNs under infinite servers semantics.

To speak about dynamic control, some previous questions should be answered. For example, *what to control*? According to the adopted time interpretation, flows through transitions should be controlled, both w.r.t. routing and service. Observe that this is not really new; the same strategy is used for QNs, where servers activity and routing of customers are controlled; analogously, when dealing with Forrester Diagrams, the opening of valves has to be controlled. Now the second question, *how to control*? The only idea is to control at routing points (what may be complex at no free-choices) and, eventually, to slow down the activity of transitions (servers in a station). As a last question, it should be decided *how to express* the control. Two main approaches can be considered: *multiplicative* (the speed of t is controlled as $\alpha\cdot\boldsymbol{\lambda}[t]$, with $\alpha\in(0,1)$) or *additive* (subtracting $\mathbf{u}$, the flow can go from $\mathbf{f}[t]$ to 0). It is not the moment to discuss that issue in detail, let us just say that they are "in essence" equivalent. Our choice here is to use the additive formulation. Proceeding in that way, using $\mathbf{u}$ as the *slow down control vector*, the state equation is now: $\dot{\mathbf{m}} = \mathbf{C}\cdot(\mathbf{f}(\mathbf{m}) - \mathbf{u})$, were $0 \le \mathbf{u} \le \mathbf{f}(\mathbf{m})$.

The above statement suggests two different remarks: (1) the system is *not positive* anymore in the classical (and restrictive) sense of Luenberger (1979) and Farina and Rinaldi (2000) (see (Silva & Recalde, 2003)); (2) the slowing down action is *dynamically bounded* by the actual state (marking) of the system.

Some results are already known for controllability in the previous framework. For the present purpose let us just point out the following.

Proposition 7. *If all transitions are controllable, reachability in timed models is equivalent to reachability in the underlying untimed models.*

In other words, if marking $\mathbf{m}$ is reachable in the untimed model $\langle\mathcal{N}, \mathbf{m}_0\rangle$, there exists a way of controlling the transitions for reaching it in the timed model $\langle\mathcal{N}, \mathbf{m}_0, \mathbf{R}, \boldsymbol{\lambda}\rangle$.

Let us assume in the sequel that *all* the transitions are controllable.

6.4.1. Optimal steady state control

The formulation can be done in an analogous way to that used for the routing optimization.

Problem 8. Maximize $\mathbf{g} \cdot \boldsymbol{\chi} - \mathbf{w} \cdot \mathbf{m}_{ss} - \mathbf{b} \cdot \mathbf{m}_0$ with respect to the control vector $\mathbf{u}$ in steady state. The basic statement of the problem (maximize the throughput of one transition) can be presented as follows:

$$\begin{array}{l} \max\{\boldsymbol{\phi}[t_i] \\ \text{s.t}\, \boldsymbol{\mu} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma \\ \quad \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in \bullet t} \left\{ \dfrac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \right\} - \mathbf{u}[t] \quad \forall t \\ \quad \mathbf{R} \cdot \boldsymbol{\phi} = 0 \\ \quad \mathbf{C} \cdot \boldsymbol{\phi} = 0, \\ \quad \boldsymbol{\sigma}, \mathbf{u}, \geq 0\} \end{array} \tag{12}$$

According to Property 7, marking $\boldsymbol{\mu}$ should be reachable by appropriate controls, possibly in an asymptotic way. At this point it is very important to observe that the above is not a LPP. In fact, this problem is very similar to (4), and the same branch and bound strategy can be used to solve it.

The net in Fig. 4 is mono-T-semiflow, with $\mathbf{x} = 1$. Considering it without any control (i.e., the *unforced* net system) the steady state flow is $\mathbf{f}[t_1] = 0.535$. Table 6 represents which are the steady state flows depending on whether p_1 or p_5 limits t_1 and whether p_1 or p_4 limits t_2.

It can be seen that in this case the optimal value is 1.5625, and can be reached using as control $\mathbf{u} = [0.781, 0, 0, 0]$.

But the above computed control is *open* loop. One very classical way for using it in a structure with feedback is to use a mixed feedforward–feedback schema (see Fig. 5). The computation of "good" controllers for the control loop requires still more attention. Nevertheless, in practice, the use of the controller that corresponds to the linear system driving the application in the optimal steady state gives usually good practical results, if no big perturbation appears.

6.4.2. Minimum reachability time control

The statement is quite classical: which is the control action for bringing the net system from $\mathbf{m}_0$ to $\mathbf{m}_f$ in minimum time? For simplicity, *let us assume that there are no immediate transitions*.

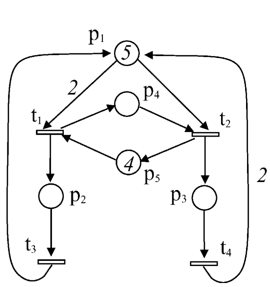

Fig. 4. In this net with $\boldsymbol{\lambda} = [3, 1, 1, 10]$, the steady state throughput of the continuous system is 0.535, while it is 0.801 as discrete.

Table 6
Best steady state control and flow depending on which is the control place of t_1 and t_2

Limiting places		$\mathbf{u}$	$\mathbf{f}[t_1]$
t_1	t_2		
p_1	p_1	$\mathbf{u} = [0.781, 0, 0, 0]$	1.5625
p_1	p_4	$\mathbf{u} = [0.781, 0, 0, 0]$	1.5625
p_5	p_1	$\mathbf{u} = [0, 3.13, 0, 0]$	0.6522
p_5	p_4	Unfeasible	–

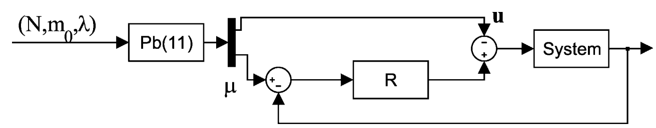

Fig. 5. Feedforward–feedback control schema.

$$\begin{array}{l} \min\{\theta \\ \text{s.t.}\, \mathbf{m}(\tau) = \mathbf{m}_0 + \int_0^\tau \mathbf{C} \cdot \mathbf{f}(\psi)\, \mathrm{d}\psi \\ \quad \mathbf{f}(\psi)[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in \bullet t} \left\{ \dfrac{\mathbf{m}(\psi)[p]}{\mathbf{Pre}[p,t]} \right\} - \mathbf{u}(\psi)[t], \quad \forall t \\ \quad \mathbf{m}(\psi), \mathbf{f}(\psi), \mathbf{u}(\psi) \geq 0 \\ \quad \mathbf{m}(\theta) = \mathbf{m}_f\} \end{array} \tag{13}$$

Solutions to problems of this type are not known in general, because of the simultaneous existence of minimum operators (non-differentiability) and state (marking) dependent constraints for the control variables. Nevertheless, using knowledge from the analysis of the net model, in some cases the problem can be solved using ad hoc procedures.

As an example, let us consider the net model of Fig. 4. Assume we want to obtain the control law to go from $\mathbf{m}_0$ to $\mathbf{m} = [1.56, 1.56, 0.156, 2.44]$, which as we have seen before is the optimum steady state. Place p_5 is *implicit* (Silva, Teruel, & Colom, 1998), so it can be removed without changing the behaviour of the model. The same can be said for the arc joining p_1 to t_2. Doing so, the plant of the system looses all the synchronizations, therefore it becomes linear, although with dynamic constraints on the control vector $\mathbf{u}$. The optimal evolution from $\mathbf{m}_0$ to $\mathbf{m}$ is a classical bang-bang control, for which here the only relevant thing is to decide the commutation time of t_2. In essence the optimal control consists on not constraining t_1 and t_3 ($\mathbf{u}[t_1] = \mathbf{u}[t_3] = 0$), keep t_4 closed ($\mathbf{u}[t_4] = \mathbf{f}[t_4]$), and start with t_2 closed and open it completely afterwards. Just when t_2 has to be opened has to be decided. By means of simulation it can be seen that $\tau = 0.27$ is the best option, and the time for reaching it is 0.37. Fig. 6 shows how the time increases if t_2 is open sooner or later.

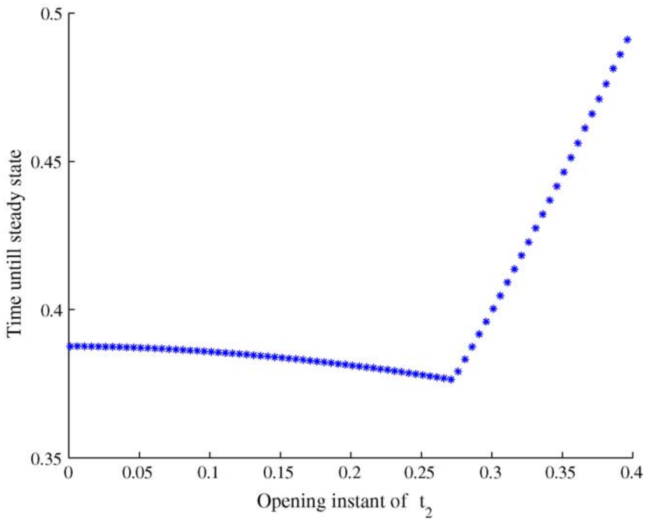

Fig. 6. Time needed to reach **m** is minimum if t_2 is opened at 0.27.

7. Some final remarks

Continuization of (time-interpreted) Petri Nets models may be a very useful relaxation technique when applied to systems with high traffic (as often happens when highly populated), independently of its nature or application domain. When the continuization is applied to all the transitions (and places), i.e., total continuization, a continuous Petri Net system is obtained. If some parts are kept discrete (partial continuization) the result is a hybrid Petri Net system. Some of the most common application domains are for example production systems (Alla & David, 1998a; Allam & Alla, 1998), inventory management systems (Giua, Furcas, Piccaluga, & Seatzu, 2001) or transportation systems (Febbraro & Sacone, 1998).

The continuization is intended as an *approximation*. In this sense, it has been observed that not every net system allows a ‘‘reasonable’’ continuization, just as not every continuous dynamic system admits to be linearized.

In general, it is expected that the study of the continuized model will be simpler than the study of the original discrete one. For example, several steady state optimization problems for continuous PNs have been introduced here which happen to be just linear programming problems.

From a more dynamic or control theory perspective, there is a lot of developments to be done, both at observation and at control levels. This is still an emerging topic, and observation of continuous Petri Net models (under infinite servers semantics) is being addressed. The development of observability criteria for continuous net systems is still a starting task (see, for example Júlvez, Jiménez, Recalde, & Silva, 2004b). For join-free nets (i.e., nets without synchronizations) a linear system is obtained (no ‘‘min’’ operator appears) and classical observability results can be applied. Structural observability (i.e., independent of firing speeds, assumed to be non-zero) is a new concept of some interest. For the design of observers, in essence observation techniques for some kind of hybrid dynamic systems are adapted to the particularities of the multilinear systems that are found for infinite servers semantics models (Júlvez, Jiménez, Recalde, & Silva, 2004a).

Analogously, at the control level, both controllability criteria and control techniques are being developed. For example, in (Júlvez, Bemporad, Recalde, & Silva 2004) hybrid control techniques based on Mixed Logic Dynamical systems (Bemporad & Morari, 1999) are applied to control continuous net systems under finite servers (constant speed) semantics.

However, the results that are obtained have to be interpreted in the original discrete model. It is necessary yet to see how to go back from the (partially) continuous relaxation to the discrete model. Moreover, not much is known for the moment about the quality of the solutions when discretized.

References

Agerwala, T., & Flynn, M. (1973). Comments on capabilities, limitations and ‘‘correctness’’ of Petri Nets. *Computer Architecture News*

Ajmone Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G., & Cumani, A. (1989). The effect of execution policies on the semantics and analysis of stochastic Petri Nets. *IEEE Transactions on Software Engineering, 15*(7), 832–846.

Ajmone Marsan, Donatelli, S., Franceschinis, G., & Neri, F. (1997). Reductions in generalized stochastic Petri nets and stochastic well-formed nets: An overview and an example of application. In J. Walrand, K. Bagchi, & G. Zobrist (Eds.), *Network performance modeling and simulation*. Gordon and Breach Publishers Inc.

Alla, H., & David, R. (1998). Continuous and hybrid Petri Nets. *Journal of Circuits, Systems and Computers, 8*(1), 159–188.

Alla, H., & David, R. (1998). A modeling and analysis tool for discrete event systems—Continuous Petri Net. *Performance Evaluation, 33*(1), 175–199.

Allam, M., & Alla, H. (1998). Modeling and simulation of an electronic component manufacturing system using hybrid Petri Nets. *IEEE Transactions on Semiconductor Manufacturing, 11*(3), 374–383.

Balbo, G., & Silva, M. (Eds.). (1998). *Proceedings of Human Capital and Mobility MATCH—Performance Advanced School*, Jaca (Spain).

Balduzzi, F., Giua, A., & Menga, G. (2000). First-order hybrid Petri Nets: A model for optimization and control. *IEEE Transactions on Robotics and Automation, 16*(4), 382–399.

Bemporad, A., Giua, A., & Seatzu, C. (2002). An iterative algorithm for the optimal control of continuous-time switched linear systems. In M. Silva, A. Giua, & J. M. Colom (Eds.), *WODES 2002: 6th Workshop on Discrete Event Systems* (pp. 335–340). Zaragoza, Spain: IEEE Computer Society.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica, 35*(3), 407–427.

Campos, J. (1998). *Performance bounds* (pp. 587–636) (in Balbo & Silva, 1998).

Campos, J., Chiola, G., & Silva, M. (1991). Ergodicity and throughput bounds of Petri Net with unique consistent firing count vector. *IEEE Transactions on Software Engineering, 17*(2), 117–125.

Campos, J., & Silva, M. (1992). Structural techniques and performance bounds of stochastic Petri Net models. In G. Rozenberg (Ed.), *Advances in Petri Nets 1992: Vol. 609. Lecture notes in computer science* (pp. 352–391). New York: Springer.

Cassandras, C. G., Sun, G., Panayiotou, C. G., & Wardi, Y. (2002). Perturbation analysis of multiclass stochastic fluid models. In *15th IFAC World Congress*.

Champagnat, R., Esteban, P., Pingaud, H., & Valette, R. (1998). Modeling and simulation of a hybrid system through Pr/Tr PN-DAE model . In *Proceeding of ADPM'98* (pp. 131–137).

Chen, H., & Yao, D. D. (2001). *Fundamentals of queueing netwroks. Performance, asymptotics and optimization: Vol. 46, applications of mathematics. Stochastic modelling and applied probability.* Springer.

Chiola, G. (1998). *Petri Nets versus queueing netwoks* (pp. 121–134) (in Balbo & Silva, 1998).

Chiola, G., Anglano, C., Campos, J., Colom, J. M., & Silva, M. (1993). Operational analysis of timed Petri Nets and application to the computation of performance bounds. In *Proceedings of the 5th International Workshop on Petri Nets and Performance Models (PNPM93)* (pp. 128–137). Toulouse, France: IEEE Computer Society Press.

Chiola, G., & Franceschinis, G. (1991). Structural colour simplification in well-formed nets. In *Proceedings of the 4th International Workshop on Petri Nets and Performance Models (PNPM91)* (pp. 20–31). IEEE Computer Society Press.

David, R., & Alla, H. (1987). Continuous Petri Nets. In *Proceedings of the 8th European Workshop on Application and Theory of Petri Nets* (pp. 275–294).

Demongodin, I., & Koussoulas, N. T. (1998). Differential Petri Nets: Representing continuous systems in a discrete-event world [Special issue]. *IEEE Transactions on Automatic Control, 43*(4), 573–579.

Dutheillet, C., Franceschinis, G., & Haddad, S. (1998). *Analysis techniques for colored well-formed systems* (pp. 233–284) (in Balbo & Silva, 1998).

Farina, L., & Rinaldi, S. (2000). *Positive linear systems. Theory and applications*. Pure and applied mathematics John Wiley and Sons.

Febbraro, A. Di., & Sacone, S. (1998). Hybrid Petri Nets for the performance analysis of transportation systems. In *Proceeding of the 37th IEEE Conference on Decision and Control* (Vol. 3, pp. 3232–3237).

Forrester, J. W. (1961). *Industrial dynamics*. MIT Press.

Forrester, J. W. (1969). *Urban dynamics*. Productivity Press.

Gaujal, B., & Giua, A. (2002). Optimal routing of continuous timed Petri Nets. In *15th IFAC World Congress*.

Giua, A., DiCesare, F., & Silva, M. (1993). Petri Net supervisors for generalized mutual exclusion constraints. In *12th IFAC World Congress* (Vol. 1, pp. 267–270).

Giua, A., Furcas, R., Piccaluga, A., & Seatzu, C. (2001). Hybrid Petri Net modeling of inventory management systems. *European Journal of Automation APII-JESA, 35*(4), 417–434.

Horton, G., Kulkarni, V. G., Nicol, D. M., & Trivedi, K. S. (1998). Fluid stochastic Petri Nets: Theory, applications, and solution techniques. *European Journal of Operational Research, 105*(4), 184–201.

Jimenez, E., Recalde, L., & Silva, M. (2001). Forrester diagrams and continuous Petri Nets: A comparative view. *Proceeding of the 8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2001),* 85–94.

Júlvez, J., Bemporad, A., Recalde, L., Silva, M. (2004). *Event-driven optimal control of continuous Petri Nets* (Research report). Dep. Informatica e Ingenieria de Sistemas, Universidad de Zaragoza. Maria de Luna, Zaragoza, Spain.

Júlvez, J., Jimenez, E., Recalde, L., & Silva, M. (2004a). *Design of observers for timed continuous Petri Net systems* (Research report). Dep. Informatica e Ingenieria de Sistemas, Univer-sidad de Zaragoza. Maria de Luna, Zaragoza, Spain.

Júlvez, J., Jimenez, E., Recalde, L., & Silva, M. (2004b). *On observability in timed continuous Petri Net systems* (Research Report). Dep. Informatica e Ingenieria de Sistemas, Univer-sidad de Zaragoza. Maria de Luna, Zaragoza, Spain.

Júlvez, J., Recalde, L., & Silva, M. (2003). On reachability in autonomous continuous Petri Net systems. In W. van der Aalst & E. Best (Eds.), *Proceedings of the 24th International Conference on Application and Theory of Petri Nets (ICATPN 2003)* (Vol. 2679, pp. 221–240). Eindhoven, The Netherlands: Springer.

Luenberger. (1979). *Introductions to dynamic systems: Theory, models and applications*. New York: John Wiley and Sons.

Mandelbau, A., & Chen, H. (1991). Discrete flow networks: Bottleneck analysis and fluid approximations. *Mathematical Operations Research, 16*(4), 408–446.

Montanari, U., & Rossi, F. (1995). Contextual nets. *Acta Informatica, 32*(4), 545–596.

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

Recalde, L., Julvez, J., & Silva, M. (2002). Steady state performance evaluation for some continuous Petri Nets. In *15th IFAC World Congress*.

Recalde, L., & Silva, M. (2001). Petri Nets fluidification revisited: Semantics and steady state. *European Journal of Automation APII-JESA, 35*(4), 435–449.

Recalde, L., Teruel, E., & Silva, M. (1998). On linear algebraic techniques for liveness analysis of P/T systems. *Journal of Circuits, Systems, and Computers, 8*(1), 223–265.

Recalde, L., Teruel, E., & Silva, M., (1999). Autonomous continuous P/T systems. In J. Kleijn & S. Donatelli (Eds.), *Application and theory of Petri Nets: Vol. 1639. Lecture notes in computer science* (pp. 107–126). Springer.

Rozenberg, G., & Reisig, W. (Eds.). (1998). *Lectures in Petri Nets, I: Basic models. Lecture notes in computer science* (Vol. 1491). New York: Springer.

Silva, M. (1993). Introducing Petri Nets. In *Practice of Petri Nets in manufacturing* (pp. 1–62). Chapman & Hall.

Silva, M., & Campos, J. (1993). Performance models based on Petri nets. In *Proceedings of the IMACS/IFAC Second International Symposium on Mathematical and Intelligent Models in System Simulation* (pp. 14–21).

Silva, M., & Recalde, L. (2002). Petri Nets and integrality relaxations: A view of continuous Petri Nets.. *IEEE Transactions on Systems, Man, and Cybernetics, 32*(4), 314–327.

Silva, M., & Recalde, L. (2003). Unforced continuous Petri Nets and positive systems. In L. Benvenuti, A. De Santis, & L. Farina (Eds.), *Positive Systems. Proceedings of the First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA 2003) of LNCIS* (Vol. 294, pp. 55–62). Italy: Springer.

Silva, M., & Teruel, E. (1996). A systems theory perspective of discrete event dynamic systems: The Petri Net paradigm. In P. Borne, J. C. Gentina, E. Craye, & S. El Khattabi (Eds.), *Symposium on discrete events and manufacturing systems. CESA '96 IMACS Multiconference* (pp. 1–12).

Silva, M., Teruel, E., & Colom, J. M. (1998). *Linear algebraic and linear programming techniques for the analysis of net systems* (pp. 309–373) (in Rozenberg & Reisig, 1998).

Teruel, E., & Silva, M. (1996). Structure theory of equal conflict systems. *Theoretical Computer Science, 153*(1–2), 271–300.

Trivedi, K., & Kulkarni, V. G. (1993). FSPNs: Fluid Stochastic Petri Nets. In M. Ajmone Marsan (Ed.), *Application theory of Petri Nets 1993: Vol. 691. Lecture notes in computer science* (pp. 24–31). Springer.

Valmari, A. (1998). In G. Rozenberg & W. Reisig (Eds.), *The state explosion problem* (pp. 429–528).

Vernon, M., Zahorjan, J., & Lazowska, E. D. (1987). A comparison of performance Petri Nets and queueing network models. In *Third International Workshop on Modelling Techniques and Performance Evaluation*.

Manuel Silva received the Industrial–Chemical Engineering degree from the University of Sevilla in 1974 and the postgraduate and Ph.D. degrees in Control Engineering from the Institut National Polytechnique de Grenoble in 1975 and 1978, respectively. From 1975 to 1978, he worked for the Centre National de la Recherche Scientifique at the Laboratoire d'Automatique de Grenoble. In 1978 he started the group of Systems Engineering and Computer Science at the University of Zaragoza, where he was named Professor in Systems Engineering and Automatic Control in 1982. His research interests include modeling, validation, performance evaluation, and implementation of distributed concurrent systems using Petri Nets, binary decision graphs, and robots programming and control. He is author of the book *Las Redes de Petri en la Automática y la Informática* (AC, 1985), coauthor of the book *Practice of Petri Nets in Manufacturing* (Chapman & Hall, 1993), and co-editor of one special issue on CIM of the IEEE Transactions on Robotics and Automation. Prof. Silva was dean of the Centro Politécnico Superior, University of Zaragoza, from 1986 to 1992 and president of the Aragonese Research Council (CONAI) and of the Research and Innovation Committee of the French–Spanish Comisión de Trabajo de los Pirineos (CTP) from 1993 to 1995. He has been associate editor of the IEEE Transactions on Robotics and Automation and associate editor of the European Journal of Control. He is advisory member of the IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences, member of the Steering Committees of the International Conference on Application and Theory of Petri Nets and of the Workshops on Discrete Event Systems, founder member of the Asociación Española de Robótica, and member of the IEEE, IFAC and EATCS. Interested in the History of Technology and Engineering, he has been distinguished with a medal from Lille and by the Association of Telecommunication Engineers of Aragón. He is member of the Royal Academy of Engineering of Spain.

Laura Recalde received the M.S. degree in Mathematics and the Ph.D. degree in Computer Engineering from the University of Zaragoza in 1995 and 1998, respectively. She is currently Assistant Professor in the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, where she is in charge of courses in Systems Theory and Discrete Event Systems. Her main research interest is the modelling and analysis of distributed concurrent systems using Petri Nets.