ON FLUIDIFICATION OF PETRI NETS: FROM DISCRETE TO HYBRID AND CONTINUOUS MODELS

Manuel Silva and **Laura Recalde** [1]

Departamento de Informática e Ingeniería de Sistemas,
Centro Politécnico Superior, Universidad de Zaragoza,
María de Luna 3, E-50018 Zaragoza, Spain

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 in order 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.

Keywords: Petri nets, relaxation, hybrid models

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, for example (Silva and 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 (David and Alla, 1987; Alla and David, 1998*a*), but there exist alternative hybrid PN formalisms around the idea of fluidification. Trivedi and his group introduced the so called *Fluid Petri Nets* (Trivedi and Kulkarni, 1993; Horton *et al.*, 1998). However, from a technical point of view the models are quite different. First let us point out that Trivedi and his colleagues deal with hybrid models, not with fluid ones: only the marking of one (or a few)

[1] Partially supported by the CICYT-FEDER project TIC2001-1819

places is relaxed into the non-negative reals. Moreover, and contrary to the approach presented here, autonomous (untimed) models are not considered. Moreover, 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 his group in (Champagnat *et al.*, 1998), or reinterpretation as in (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 (for example, 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). Moreover, it happens that continuous models obtained from Petri Nets are *in essence hybrids*, more precisely *piecewise linear* in the basic cases of *infinite* and *single servers semantics* (Silva and 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 studied 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 (Dicesare *et al.*, 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 $\mathbf{Pre}$ and $\mathbf{Post}$ 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 $p \in {}^\bullet t$, $\mathbf{m}[p] > \mathbf{Pre}[p,t]$, and its *enabling degree* is $\mathrm{enab}(t, \mathbf{m}) = \lfloor \min_{p \in {}^\bullet t} \{\mathbf{m}[p]/\mathbf{Pre}[p,t]\} \rfloor$. The firing of t in a certain amount $\alpha \in \mathbb{N}$, $\alpha \leq \mathrm{enab}(t, \mathbf{m})$ 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*. If $\mathbf{m}$ is reachable from $\mathbf{m_0}$ through a sequence σ, a *state (or fundamental) equation* can be written: $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}$ is the firing count vector.

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 $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$, $\mathbf{x} > \mathbf{0}$ it is said to be *consistent*.

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

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 $\mathbf{R}$. 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 $r_i / (\sum_{j=1}^{k} r_j)$.

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* (for example, reachability analysis, 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 an 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). 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, errors due to the relaxation of timed models happen to be not very significant in practice when relatively heavy traffic conditions are relaxed.

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

Queuing Networks (QN) are 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, for example, (A. Mandelbau and, 1991; Chen and Yao, 2001; Cassandras *et al.*, 2002)). Forrester Diagrams, appeared in the Systems Dynamics framework, is a well-known, essentially continuous, formalism for modelling certain classes of DES (Forrester, 1961; Forrester, 1969). Comparisons of PNs and (monoclass) QNs or FDs can be seen in (Vernon *et al.*, 1987; Silva and Campos, 1993; Chiola, 1998), and (Jiménez *et al.*, 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 an 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 absence of local (material, energy, information) conservation rules 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*. Nevertheless if all the transitions are continuized the net is said to be *"continuous"* (Alla and David, 1998*a*; Recalde and 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 and Rossi, 1995), an extension of net models that escape from the consumption/production logic, but that sometimes are "inaccurately" modelled by self-loops (what 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 and 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 reading 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 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. This way 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.

In timed models, in order to associate a time semantics to the fluidification of a transition, it should be taken into account that a transition is like an 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.

For the timing interpretation of continuous and hybrid PNs we will use a first order (or deterministic) approximation of the discrete case (Recalde and 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 *et al.*, 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 and David, 1998*a*; Recalde and Silva, 2001). Under finite servers semantics, the flow of t_i has just an upper bound, $\boldsymbol{\lambda}[t_i]$ (the number of servers times the speed of a server). Then $\mathbf{f}(\tau)[t_i] \leq \boldsymbol{\lambda}[t_i]$ (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, i.e., $\mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \mathrm{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p,t]\}$, leading to piecewise linear differential systems. As in discrete PNs, immediate transitions act as flow splitters. The system obtained adding this equation to the state equation is positive (Luenberger, 1979; Farina and Rinaldi, 2000), that is, $\mathbf{m} \geq \mathbf{0}$ is redundant.

It should be pointed out that finite server semantics, equationally modelled by bounding the firing speed of continuized transitions, corresponds at 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). On the other hand, infinite servers semantics really relax 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 *et al.*, 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 Coloured PN models (Dutheillet *et al.*, 1998; Chiola and Franceschinis, 1991; Ajmone Marsan and Neri, 1997), a product of state variables appears (as for the classical Volterra-Lotka model for a basic population dynamic case) (Silva and 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 infinite servers semantics.

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 and Silva, 1992). Let χ_j be the throughput of transition t_j (i.e., the number of firings per time unit). The throughput vector of the net system is $\boldsymbol{\chi} = [\chi_j]$. 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: $\mathbf{v}^{(i)}[t_j] = \boldsymbol{\chi}[t_j]/\boldsymbol{\chi}[t_i]$. 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

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)

Table 2. 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 reducible nets	$\mathbf{v}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{R})$
Simple nets	$\mathbf{v}(\mathcal{N}, \mathbf{m_0}, \mathbf{R}, \boldsymbol{\lambda})$

hierarchy of nets, and some well-known net subclasses are re-encountered (see Table 2).

In particular, *mono T-semiflow reducible* nets are defined as those PNs for which the following system has a unique solution:

$$\mathbf{C} \cdot \mathbf{v}^{(1)} = \mathbf{0}$$

$$\mathbf{R} \cdot \mathbf{v}^{(1)} = \mathbf{0} \quad \text{for the resolution of EQ conflicts among } \textit{immediate} \text{ transitions}$$

$$\frac{\mathbf{v}^{(1)}[t_i]}{\boldsymbol{\lambda}[t_i]} = \frac{\mathbf{v}^{(1)}[t_j]}{\boldsymbol{\lambda}[t_j]} \quad \forall\, t_i, t_j \;\; \textit{timed} \text{ transitions in EQ relation}$$

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

For example, str. live and bounded EQ nets with conflicts solved through immediate transitions belong to the class. Observe that in this class the visit ratio only depends on $\mathcal{N}$, $\mathbf{R}$ and $\boldsymbol{\lambda}$, i.e., the structure and interpretation, but not on $\mathbf{m_0}$.

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). 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 (see Figure 1), it can be said that transitions with relatively high visit ratio are "good" candidates for fluidification.

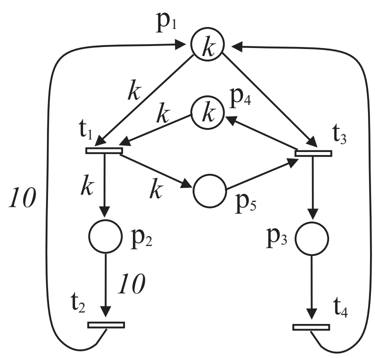

Fig. 1. The unique T-semiflow is $\mathbf{x} = [1, k/10, k, k]$, thus "a priori" the less reasonable candidate to fluidify is t_1

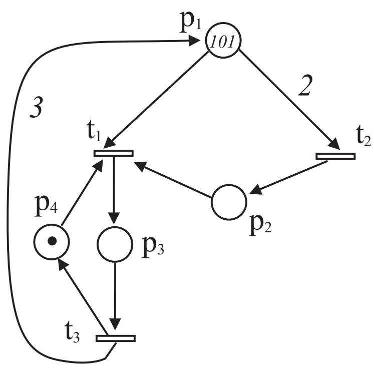

Fig. 2. A system that is live as discrete, but non live as continuous or hybrid (with t_2 as continuous)

4. PARTIAL CONTINUIZATION IS NOT ALWAYS POSSIBLE

Not all Petri net systems allow such continuization as an approximate modelling. Some important technical problems may appear.

For example, the autonomous system in Figure 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.

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 Figure 1, with firing speed vector $\boldsymbol{\lambda} = [1, 3, 30, 30]$, under infinite servers semantics. Taking into account that transition t_1 will be fired far less than the others and 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 have exponential pdf (0.602) or are defined as deterministic (0.714). 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 3). 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 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 *et al.*, 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 *et al.*, n.d.).

5.1 Autonomous models

The fundamental equation 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 (Recalde *et al.*, 1999; Júlvez *et al.*, n.d.), and starts increasing 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 *et al.*, 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.*, 1998; Recalde *et al.*, 1999)

- Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a live and bounded system. Then, $\mathcal{N}$ is consistent, conservative and $\text{rank}(\mathbf{C}) \leq |\text{SEQS}| - 1$.
- Let $\mathcal{N}$ be a consistent and conservative net. If $\text{rank}(\mathbf{C}) = |\text{SCCS}| - 1$, then $\mathcal{N}$ is str. live as discrete.

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

Theorem 3. (Teruel and Silva, 1996; Recalde *et al.*, 1999)

An EQ net is str. live and str. bounded if and only if it is consistent, conservative and $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$.

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

Table 3. Throughput differences in the system in Figure 1

	Single server	Infinite servers
Discrete	0.423 (Det.), 0.677 (Mark.)	0.714 (Det.), 0.6 (Mark.)
Continuous	1	0.73
Hybrid	0.5	0

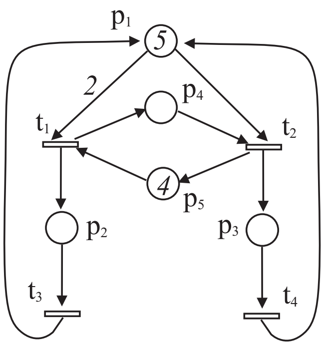

Fig. 3. In this net with $\boldsymbol{\lambda} = [3, 1, 1, 10]$, the steady state throughput of the continuous system is 0.535, while it is 0.801 as discrete.

continuous or hybrid case, since they are structural objects. However, 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 and David, 1998*b*) 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 Figure 3).

For *discrete* net systems an upper bound of the throughput ($\boldsymbol{\phi} \geq \boldsymbol{\chi}$) can be obtained by means of the following linear programming problem (Chiola *et al.*, 1993; Campos, 1998).

$$\begin{aligned} \max\{\ & \boldsymbol{\phi}[t_i] \\ \text{s.t. } & \boldsymbol{\mu} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^{\bullet}t \\ & \mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0}, \\ & \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\phi} \geq \mathbf{0}\} \end{aligned} \tag{1}$$

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 $\mathbf{C} \cdot \boldsymbol{\phi} \geq \mathbf{0}$ if the net is not structurally bounded). The linear programming problem is exact (that is, $\boldsymbol{\chi} = \boldsymbol{\phi}$) 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 as follows: $\boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]}$ if $p = {}^{\bullet}t$.

Unfortunately, in general it is not a bound of the system, just an approximation, although it is exact for EQ nets.

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

$$\begin{aligned} \max\{\ & \boldsymbol{\phi}[t_i] \\ \text{s.t. } & \boldsymbol{\mu} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^{\bullet}t} \left\{ \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \right\} \forall t \in T \\ & \mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\phi} \geq \mathbf{0}\} \end{aligned} \tag{2}$$

A branch and bound algorithm can be used to solve this programming problem. The idea is to solve the relaxed LPP defined in (1). 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.

In mono-T-semiflow reducible nets the solution of system (2) is an upper bound of the steady state throughput (Recalde *et al.*, 2002) of the continuous system. In this class, $\mathbf{v}^{(1)}$ is completely defined, and since $\boldsymbol{\chi}$ is proportional to $\mathbf{v}^{(1)}$, $\boldsymbol{\chi} = \alpha \cdot \mathbf{v}^{(1)}$. Hence, maximizing $\boldsymbol{\chi}[t_i]$ is equal to maximizing α. To simplify, let us assume that the system is consistent (the generalization to non consistent systems is immediate). Applying this to the LPP (1) $\boldsymbol{\phi} \geq \boldsymbol{\chi}$ can be obtained as follows:

$$\begin{aligned} \boldsymbol{\phi}[t_i] = \max\{ \ & \alpha \\ \text{s.t. } & \boldsymbol{\mu} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \alpha \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \leq \boldsymbol{\mu} \\ & \boldsymbol{\mu} \geq \mathbf{0} \} \end{aligned} \tag{3}$$

where $\mathbf{D}^{(i)}$ is the vector of *average service demands*, ($\mathbf{D}^{(i)}[t] = \mathbf{v}^{(i)}[t] \cdot \mathbf{s}[t]$), and $\mathbf{s}$ is the vector of mean service times, i.e., $\mathbf{s}[t] = 1/\boldsymbol{\lambda}[t]$ for the timed transitions and 0 for the immediate ones.

Applying duality and changing variables, the following classical LPP can be written (Chiola *et al.*, 1993):

$$\begin{aligned} 1/\boldsymbol{\phi}[t_i] = \max\{ \ & \mathbf{y} \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \\ \text{s.t. } & \mathbf{y} \cdot \mathbf{C} = \mathbf{0} \\ & \mathbf{y} \cdot \mathbf{m_0} = 1 \\ & \mathbf{y} \geq \mathbf{0} \} \end{aligned} \tag{4}$$

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 logically constrained by the required controller (*generalized mutual exclusion controller*, GMEC (Giua *et al.*, 1993), *supervisory controller* (Ramadge and Wonham, 1989), ...).

6.1 Design problems

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 6.2.

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 ($\boldsymbol{\mu}_0$), 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 (for example, 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 try to *minimize a cost function* (for example 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 equal conflict (EQ) nets (its characterization can be computed polynomially through the rank theorem, see Section 5.1). Let us review 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, and conflicts are solved according to routing rates,* $\mathbf{R}$.

Problem 4. (Silva and 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 $\mathbf{g} = \mathbf{1}_i, \mathbf{V} = \mathbf{0}$ and $\mathbf{b} = \mathbf{0}$).

Simply adding the cost constraints to LPP (3), the following can be written:

$$\begin{aligned}\mathbf{\Gamma}[t_i] = \max\{\ & \alpha \\ \text{s.t. } & \alpha \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \leq \boldsymbol{\mu}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \mathbf{V} \cdot \boldsymbol{\mu}_0 \leq \mathbf{k} \\ & \boldsymbol{\mu}_0 \geq \mathbf{0}\}\end{aligned} \tag{5}$$

If LPP (1) is used instead as the basis (it is equivalent if the visit ratio is computable with $\mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0}$ and $\mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0}$) (Campos, 1998), 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:

$$\begin{aligned}\max\{\ & \boldsymbol{\phi}[t_i] \\ \text{s.t. } & \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^{\bullet}t \\ & \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \text{if } p = {}^{\bullet}t \\ & \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \boldsymbol{\mu}, \boldsymbol{\mu}_0 \geq \mathbf{0} \\ & \mathbf{V} \cdot \boldsymbol{\mu}_0 \leq \mathbf{k}\}\end{aligned} \tag{6}$$

Problem 5. (Silva and Recalde, 2002)

Given a cost weighting $\mathbf{b}$ and a cycle time $\Gamma_i = 1/\boldsymbol{\phi}_i$, compute the minimal cost initial marking, $\mathbf{b} \cdot \boldsymbol{\mu}_0$, to have the given cycle time:

$$\begin{aligned}\min\{\ & \mathbf{b} \cdot \mathbf{m_0} \\ \text{s.t. } & \mathbf{C} \cdot \mathbf{z} + \Gamma_i \cdot \boldsymbol{\mu}_0 \geq \mathbf{Pre} \cdot \mathbf{D}^{(i)} \\ & \boldsymbol{\mu}_0 \geq \mathbf{0}\}\end{aligned} \tag{7}$$

Alternatively, if LPP (1) 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*.

$$\begin{aligned}\min\{\ & \mathbf{b} \cdot \boldsymbol{\mu}_0 \\ \text{s.t. } & \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^{\bullet}t \\ & \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \text{if } p = {}^{\bullet}t \\ & \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \boldsymbol{\mu}, \boldsymbol{\mu}_0 \geq \mathbf{0} \\ & \boldsymbol{\phi}[t_i] = 1/\Gamma_i\}\end{aligned} \tag{8}$$

Optimization of routing, $\mathbf{R}$. 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 str. live and bounded EQ system.*

Problem 6. Maximize $\mathbf{g} \cdot \boldsymbol{\phi} - \mathbf{w} \cdot \boldsymbol{\mu} - \mathbf{b} \cdot \mathbf{m_0}$, 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 (1) has been to *remove* the constraint on the routing.

$$\begin{aligned}\max\{\ & \mathbf{g} \cdot \boldsymbol{\phi} - \mathbf{w} \cdot \boldsymbol{\mu} - \mathbf{b} \cdot \mathbf{m_0} \\ \text{s.t. } & \boldsymbol{\mu} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\phi}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^{\bullet}t \\ & \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \quad \text{if } p = {}^{\bullet}t \\ & \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0} \\ & \boldsymbol{\mu} \geq \mathbf{0}\}\end{aligned} \tag{9}$$

Once LPP (9) has been solved, the computation of the routing matrix $\mathbf{R}$, is straightforward, just proceeding free-choice by free-choice. Assuming for simplicity that choices are binary: $\boldsymbol{\phi}_1/\boldsymbol{\phi}_2 = r_1/r_2$ (for example, take $r_1 + r_2 = 1$).

If all conflicts are solved with immediate transitions, this LPP is equivalent to the one stated in (Gaujal and Giua., 2002).

6.2 Control problems

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 *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.

In order to speak about 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 for routing and service. Observe that this is nothing new; the same strategy is used for QNs, were 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 $\mathbf{0}$). It is not the moment to discuss that issue in detail, let us just say that they are "in essence" equivalent. Nevertheless, our choice here is to use the additive formulation. Proceeding in that way, using $\mathbf{u}$ as the *slow down control vector*, the set of equations is: $\dot{\mathbf{m}} = \mathbf{C} \cdot (\mathbf{f}(\mathbf{m}) - \mathbf{u})$, were $\mathbf{0} \leq \mathbf{u} \leq \mathbf{f}(\mathbf{m})$.

The above statement suggests two different remarks: (1) the system is *not positive* anymore in the classical sense of (Luenberger, 1979; Farina and Rinaldi, 2000); (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.

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

Problem 8. Maximize $\mathbf{g} \cdot \boldsymbol{\phi} - \mathbf{w} \cdot \boldsymbol{\mu} - \mathbf{b} \cdot \mathbf{m_0}$ with respect to the control vector $\mathbf{u}$ in steady state.

The basic statement of the problem can be presented as follows:

$$
\begin{aligned}
\max\{\ & \boldsymbol{\phi}[t_i] \\
\text{s.t}\ & \boldsymbol{\mu} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\
& \boldsymbol{\phi}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p,t]} \right\} - \mathbf{u}[t] \ \ \forall t \\
& \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0}, \\
& \mathbf{R} \cdot \boldsymbol{\phi} = \mathbf{0} \\
& \boldsymbol{\mu}, \mathbf{u}, \geq \mathbf{0} \}
\end{aligned} \tag{10}
$$

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. Like for the computation of the flow in mono T-semiflow reducible nets (Recalde *et al.*, 2002), some tree based computations are needed. The net in Figure 3 is mono-T-semiflow, with $\mathbf{x} = \mathbf{1}$. Considering it without any control (i.e., the *unforced* net system) the steady state flow is $\mathbf{f}[t_1] = 0.535$. Table 4 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.

Table 4. Best steady state control and flow depending on which is the control place of t_1 and t_2

Limiting places t_1	t_2	$\mathbf{u}$	$\mathbf{f}[t_1]$
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	-

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 an structure with feedback is to use a mixed feedforward-feedback schema (see Figure 4). 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.

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.*

$$
\begin{aligned}
\min\{\ & \theta \\
\text{s.t.}\ & \mathbf{m}(\tau) = \mathbf{m_0} + \int_0^\tau \mathbf{C} \cdot \mathbf{f}(\psi) d\psi \\
& \mathbf{f}(\tau)[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\mathbf{m}(\tau)[p]}{\mathbf{Pre}[p,t]} \right\} - \mathbf{u}(\tau)[t] \ \forall t \\
& \mathbf{m}(\tau), \mathbf{f}(\tau), \mathbf{u}(\tau) \geq \mathbf{0} \\
& \mathbf{m}(\theta) = \mathbf{m}_f \}
\end{aligned} \tag{11}
$$

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 Figure 3. 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 *et al.*, 1998), so it can be removed without changing the

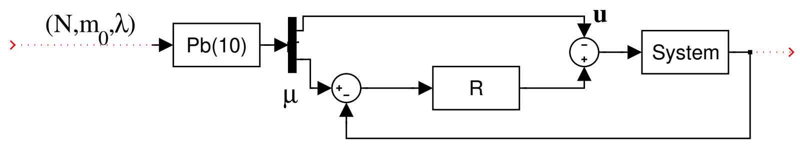

Fig. 4. Feedforward-feedback control schema

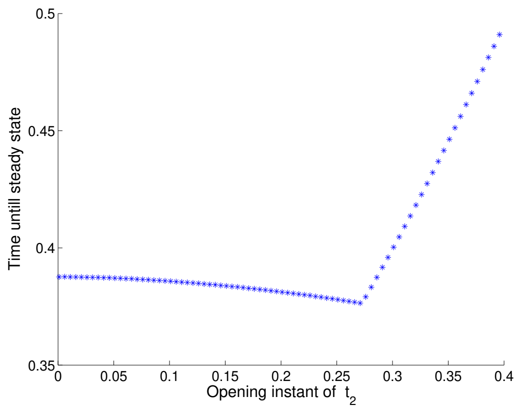

Fig. 5. Time needed to reach $\mathbf{m}$ is minimum if t_2 is opened at 0.27

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. Figure 6.2 shows how the time increases if t_2 is open sooner or later.

REFERENCES

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

Agerwala, T. and M. Flynn (1973). Comments on capabilities, limitations and "correctness" of Petri nets. *Computer Architecture News*.

Ajmone Marsan, S Donatelli, G. Franceschinis and F. Neri (1997). Reductions in generalized stochastic petri nets and stochastic well-formed nets: An overview and an example of application. In: *Network Performance Modeling and Simulation* (J. Walrand, K. Bagchi and G. Zobrist, Eds.). Gordon and Breach Publishers INC.

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

Alla, H. and R. David (1998*b*). A modeling and analysis tool for discrete event systems – continuous petri net.. *Performance Evaluation* **33**(3), 175–199.

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

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

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

Campos, J. (1998). Performance bounds. Chap. 17, pp. 587–636. In: Balbo and Silva (1998).

Campos, J. and M. Silva (1992). Structural techniques and performance bounds of stochastic Petri net models. In: *Advances in Petri Nets 1992* (G. Rozenberg, Ed.). Vol. 609 of *Lecture Notes in Computer Science*. pp. 352–391. Springer.

Cassandras, C. G., G. Sun, C. G. Panayiotou and Y. Wardi (2002). Perturbation analysis of multiclass stochastic fluid models. In: 15^{th} *IFAC World Congress*. Barcelona, Spain.

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

Chen, H. and D.D. Yao (2001). *Fundamentals of Queueing Netwroks. Performance, Asymptotics and Optimization*. Vol. 46 of *Applications of Mathematics. Stochastic Modelling and Applied Probability*. Springer.

Chiola, G. (1998). Petri nets versus queueing netwoks. Chap. 4, pp. 121–134. In: Balbo and Silva (1998).

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

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

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

Demongodin, I. and N. T. Koussoulas (1998). Differential Petri nets: Representing continuous systems in a discrete-event world. *IEEE Trans. on Automatic Control (Special Issue)* **43**(4), 573–579.

Dicesare, F., G. Harhalakis, J. M. Proth, M. Silva and F. B. Vernadat (1993). *Practice of Petri Nets in Manufacturing*. Chapman & Hall.

Dutheillet, C., G. Franceschinis and S. Haddad (1998). Analysis techniques for colored well-formed systems. Chap. 7, pp. 233–284. In: Balbo and Silva (1998).

Farina, L. and S. Rinaldi (2000). *Positive Linear Systems. Theory and Applications*. Pure and Applied Mathematics. John Wiley and Sons. New York.

Forrester, Jay W. (1961). *Industrial Dynamics*. MIT Press. Cambridge, Mass.

Forrester, Jay W. (1969). *Urban Dynamics*. Productivity Press.

Gaujal, B. and A. Giua. (2002). Optimal routing of continuous timed petri nets. In: *15^{th} IFAC World Congress*. Barcelona, Spain.

Giua, A., F. DiCesare and M. Silva (1993). Petri net supervisors for generalized mutual exclusion constraints. In: 12^{th} *IFAC World Congress*. Vol. 1. Sidney, Australia. pp. 267–270.

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

Jiménez, E., L. Recalde and M. Silva (2001). Forrester diagrams and continuous Petri nets: A comparative view. In: *Proc. of the 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2001)*. pp. 85–94.

Júlvez, J., L. Recalde and M. Silva (n.d.). On reachability in autonomous continuous petri net systems. Accepted at the 24th Int. Conf. on Application and Theory of Petri Nets.

Luenberger (1979). *Introductions to Dynamic Systems: Theory, Models and Applications*. John Wiley and Sons. New York.

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

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

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

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

Recalde, L., E. Teruel and M. Silva (1999). Autonomous continuous P/T systems. In: *Application and Theory of Petri Nets 1999* (J. Kleijn S. Donatelli, Ed.). Vol. 1639 of *Lecture Notes in Computer Science*. Springer. pp. 107–126.

Recalde, L., J. Júlvez and M. Silva (2002). Steady state performance evaluation for some continuous petri nets. In: *15^{th} IFAC World Congress*. Barcelona, Spain.

Rozenberg, G. and Reisig, W., Eds.) (1998). *Lectures in Petri Nets, I: Basic Models*. Vol. 1491 of *Lecture Notes in Computer Science*. Springer.

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

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

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

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

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

Trivedi, K. and V. G. Kulkarni (1993). FSPNs: Fluid stochastic Petri nets. In: *Application and Theory of Petri Nets 1993* (M. Ajmone Marsan, Ed.). Vol. 691 of *Lecture Notes in Computer Science*. pp. 24–31. Springer.

Valmari, A. (1998). The state explosion problem. pp. 429–528. In: Rozenberg and Reisig (1998).

Vernon, M., J. Zahorjan and E. D. Lazowska (1987). A comparison of performance Petri nets and queueing network models. In: *3rd Int. Workshop on Modelling Techniques and Performance Evaluation*. Paris, France.