

# Continuization of Timed Petri Nets: From Performance Evaluation to Observation and Control

Manuel Silva and Laura Recalde \*

Dep. Informática e Ingeniería de Sistemas, Centro Politécnico Superior de Ingenieros,  
Universidad de Zaragoza, María de Luna 3, E-50012 Zaragoza, Spain  
`silva,lrecalde@unizar.es`

**Abstract.** State explosion is a fundamental problem in the analysis and synthesis of discrete event systems. Continuous Petri nets can be seen as a relaxation of discrete models allowing more efficient (in some cases polynomial time) analysis and synthesis algorithms. Nevertheless computational costs can be reduced at the expense of the analyzability of some properties. Even more, some net systems do not allow any kind of continuization. The present work first considers these aspects and some of the alternative formalisms usable for continuous relaxations of discrete systems. Particular emphasis is done later on the presentation of some results concerning performance evaluation, parametric design and marking (i.e., state) observation and control. Even if a significant amount of results are available today for continuous net systems, many essential issues are still not solved. A list of some of these are given in the introduction as an invitation to work on them.

## 1 Introduction

It is well-known that Discrete Event Dynamic Systems (DEDS), and in particular Petri Nets (PN), suffer the *state explosion problem*, what is particularly true when the system is “highly populated” (i.e., the initial marking is large). One way to tackle that problem is to use some kind of *relaxation*. *Fluidification* (or *continuization*) is a classical relaxation technique that tries to deal with the state explosion by removing the integrality constraints. The idea is analogous to that allowing the transformation of an Integer Linear Programming problem (ILP, NP-hard) into a Linear Programming problem (LP, polynomial time complexity). The systematic use of linear programming in the fundamental or “state” equation for the analysis of Petri Nets was proposed in 1987 (see [1] for a revised version of the seminal work, and [2] for a more recent survey).

In Queuing Networks (QN) approximating the clients flow with a continuous fluid flow is a classical relaxation (see, for example, [3–6]). For PN, a similar relaxation was introduced and developed by R. David and co-authors, starting

---

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

in 1987 [7] (see [8] for a very recent perspective). These models were called *continuous Petri Nets*.

Fluidified models have advantages in the sense that they allow to obtain better analytical characterizations, or computationally more efficient algorithms. However, being an approximation, there are properties that cannot be analyzed (mutex for example), and often only partial results can be obtained with respect to their validity in the original system [9].

Even if the idea of continuization of Petri nets is well inscribed in the framework of classical relaxations, even if several interesting analysis and synthesis results have been obtained in the last years [8–11], the field is still very young and essential contributions are needed. Let us just start this overview —we recognize partially biased from and towards our works— saying that many essential questions do not have a satisfactory answer, in several cases because the problem has not been addressed yet. Just quoting a few:

1. Is a given net system fluidizable? Some net models are not “approximated” at all by its fluidization, just as many differential equation systems (as chaotic models) do not admit a reasonable linearization.
2. How to define the firing policies of transitions when submerging nets systems in time (i.e., how to define the *routing decisions* and *service rates* at conflicts and at stations, respectively)? Today two servers semantics (finite or constant speed, and infinite or variable speed) are mostly used in a *deterministic* setting, but many others can be defined.
3. Provided that several firing (service and routing) semantics can be defined, which one is the best —or a good one— for a given particular case?
4. Given a timing semantics: When does a steady state exist? (The apparent mismatch being that the underlying markovian model may be ergodic, while the continuous case is oscillatory.)
5. Marking reachability in untimed net systems has today a quite reasonable characterization in algebraic terms. For timed models, even for steady state markings, only necessary conditions are well known in general. Of course, for particular classes of net systems (for example, live and bounded equal conflicts), more powerful results are available. The issue is: How to improve the characterization of which steady states can be reached from a given initial marking (and, eventually, which is a “good” control policy)?
6. Even if for *off-line design* problems some interesting results are already known, *observation* and, essentially, *control* of continuous net models require still important improvements. Lose of observability or stability requires still much work.
7. Assuming that “good” off-line designs or dynamic controls are obtained for the continuous relaxation, how to come back to a “reasonable” design or control (scheduling) in the original discrete setting? For this problem, some post-optimization strategies (eventually using metaheuristics like simulated annealing or taboo search [12] can be used, but the problem is essentially unexplored).

Therefore in this work no concluding remarks will be given, leaving the presentation relatively open. The reader is recommended to have the above open questions and many others in mind while going through the following material. Even if the warning to existing “holes” in our basic understanding (the theory) is put here in the introduction, some hopefully interesting results are available. They provide some behavioral characterizations, sometimes in polynomial time.

The structure of the paper is as follows: In Sect. 2 both autonomous and timed continuous PN are introduced. Sect. 3 briefly compares PN with other formalisms in which similar relaxations are used (queuing networks, Forrester diagrams and positive systems). Performance evaluation of continuous timed PN is addressed in Sect. 4. The results that are obtained are applied in Sect. 5 to some synthesis problems. Sect. 6 is devoted to the study of observability. Finally, the dynamic control of continuous timed models is considered in Section 7.

## 2 Continuous Petri Nets: on the Relaxation of DEDS Models

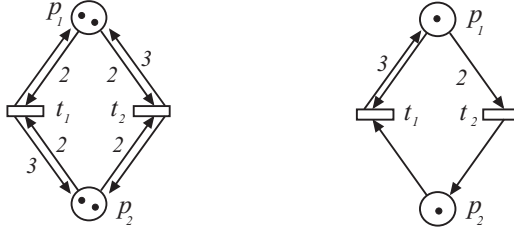
### 2.1 Autonomous Continuous Petri Nets

We assume that the reader is familiar with PN (for notation we use the standard one, see for instance [13]).

The usual PN system,  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ , will be said to be *discrete* so as to distinguish it from a *continuous* PN. The structure  $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$  of continuous Petri nets is the same as in discrete PN, the difference is in the evolution rule. In continuous PN firing is not restricted to be done in integer amounts. As a consequence the marking is not forced to be integer. More precisely, a transition  $t$  is *enabled* at  $\mathbf{m}$  iff for every  $p \in \bullet t$ ,  $\mathbf{m}[p] > 0$ , and its *enabling degree* is  $\text{enab}(t, \mathbf{m}) = \min_{p \in \bullet t} \{ \mathbf{m}[p] / \mathbf{Pre}[p, t] \}$ . The firing of  $t$  in a certain amount  $\alpha \leq \text{enab}(t, \mathbf{m})$  leads to a new marking  $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}[P, t]$ .

The set of reachable markings in continuous PN verifies some properties that do not hold in the discrete case. For example, the set of reachable markings of a continuous system is a convex set [14].

In continuous Petri nets the firing can be done in so small quantities that a net can be “almost” in a deadlock state but never reach it. In our opinion, these *limit markings* should also be considered as reachable. Otherwise, there are nets, as the one in Fig. 1(a), that will deadlock as discrete for any initial marking, but can never be *completely* blocked as continuous. In [14] the limit reachability concept was introduced. The set of these limit-reachable markings will be denoted as  $\text{RS}_C(\mathcal{N}, \mathbf{m}_0)$ . The set of solutions of the relaxed fundamental equation will be denoted as  $\text{LRS}_C(\mathcal{N}, \mathbf{m}_0)$ , that is,  $\text{LRS}_C(\mathcal{N}, \mathbf{m}_0) = \{ \mathbf{m} \mid \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0}, \boldsymbol{\sigma} \geq \mathbf{0} \}$ . As in discrete PN, the fundamental equation relaxation may add spurious solutions to the relaxation made at the net level, that is,  $\text{RS}_C(\mathcal{N}, \mathbf{m}_0) \subseteq \text{LRS}_C(\mathcal{N}, \mathbf{m}_0)$ . However, and contrary to what happens in the discrete case, in most practical cases these sets are equal. More precisely, if  $\mathcal{N}$  is consistent (i.e.,  $\exists \mathbf{x} > \mathbf{0} : \mathbf{C} \cdot \mathbf{x} = \mathbf{0}$ ) and every transition can



**Fig. 1.** (a) If limit markings are not considered, liveness of a continuous net is not sufficient for (structural) liveness of its discrete counterpart. (b) In any case, (structural) liveness of the continuous net is not necessary for (structural) liveness of its discrete counterpart.

be fired,  $\exists \sigma > \mathbf{0}, \mathbf{m}_0 \xrightarrow{\sigma}$  (or, equivalently,  $\exists \mathbf{m} > \mathbf{0}, \mathbf{m} \in \text{RS}_C(\mathcal{N}, \mathbf{m}_0)$ ), then:  $\text{RS}_C(\mathcal{N}, \mathbf{m}_0) = \text{LRS}_C(\mathcal{N}, \mathbf{m}_0)$  [14]. Since the preconditions are very weak, this means that *in practice* the relaxation at net level is “equivalent” to the relaxation at the fundamental equation level. That is, there do not exist *spurious* solutions of the fundamental equation.

Hence, properties like deadlock-freeness can be analytically studied. Moreover, structural traps ( $\Theta \subseteq P$  such that  $\Theta^\bullet \subseteq \bullet\Theta$ ) do not necessarily “trap” tokens in continuous PN. That is, the behavioural counterpart of structural traps is not true anymore: in continuous PN, traps may be emptied. Moreover, if every transition of the net system can be fired, then every T-semiflow can be fired in isolation [14]. This has an important consequence: *behavioural and structural synchronic relations* [15] coincide.

Fluidification means simplifying the model, assuming that this will allow to use computationally less expensive techniques and get more information about the system. However, one has to keep on mind that those results refer to the continuous PN, and do not always provide “useful” information about the underlying discrete model. Since continuous PN are a relaxation of discrete PN, for those properties based on universal (existential) quantifiers the continuous PN will provide sufficient (necessary) conditions. For example, if the continuous PN is bounded, so will be the discrete PN. For a marking to be reachable in the discrete model, reachability in the continuous one must be guaranteed. However, for those properties formulated interleaving universal and existential quantifiers the analysis of the continuous PN may not provide information about the behaviour of its discrete counterpart. For example, liveness (deadlock-freeness) of the continuous PN is neither necessary nor sufficient for liveness (deadlock-freeness) of the discrete PN. Nevertheless, to be fair one should take into account that maybe the only problem of the discrete net is that it does not have enough tokens. In fact, it can be proved that any (lim-)live continuous PN is structurally live as a discrete PN, although not necessarily live, i.e., the structure of the net is “correct”, although the marking may be “not large enough” [14]. On the other hand, a live discrete net may be so only with a particular marking, and any

increase of the marking makes it non-live (see Fig. 1(b)). That kind of nets will never go well with continuization, since continuization can be interpreted as if the marking were multiplied by a very large number (infinite populations).

Continuization leads to “easier to analyze” models compared to the discrete models. Nevertheless, the price that has been payed for the relaxation is that some properties of discrete PN cannot be observed in continuous systems, for example mutex relationship, since this property is based on the notion of disjunctive resources, which is lost in the continuous models. Also the distinction between reversibility and existence of home states is lost. This clearly extends to some monopoly and fairness situations.

## 2.2 Timed Continuous Petri Nets

A simple and interesting way to introduce time in discrete PN is to assume that all the transitions are timed with exponential probability distribution function (pdf). This way, a purely markovian performance model is obtained, for which, due to the memoryless property, the state of the underlying Markov chain is the very marking of the autonomous PN [16].

For the timing interpretation of continuous PN we will use a first order (or deterministic) approximation of the discrete case [17], assuming that the delays associated to the firing of transitions can be approximated by their mean values. Notice that for “congested” systems, this approximation is valid for any pdf — applying the central limit theorem. Here, for simplicity, *immediate* transitions will only be used for “free” conflicts that will be solved according to (marking and time independent) routing rates  $\mathbf{R}$ .

Different semantics have been defined for continuous timed transitions, the two most important being *infinite server* (or *variable speed*) and *finite server* (or *constant speed*) [18, 17]. Under finite server semantics, the flow of  $t_i$  has just an upper bound,  $\lambda[t_i]$  (the number of servers times the speed of a server). Then  $\mathbf{f}(\tau)[t_i] \leq \lambda[t_i]$  (knowing that at least one transition will be in saturation, that is, its utilization will be equal to 1). Under infinite server semantics, the flow through a timed transition  $t$  is the product of the speed,  $\lambda[t]$ , and the instantaneous enabling of the transition, i.e.,  $\mathbf{f}[t] = \lambda[t] \cdot \text{enab}(t, \mathbf{m}) = \lambda[t] \cdot \min_{p \in \bullet_i} \{\mathbf{m}[p] / \mathbf{Pre}[p, t]\}$ . In both cases piecewise linear differential systems are obtained.

For discrete PN infinite server semantics is *more general*, since it allows to implement finite server semantics by adding a place marked with as many tokens as the number of servers. However, this *does not* represent finite-servers semantics if these tokens are interpreted as fluids. In the continuous case the two evolution rules are related to different relaxations of the model. A transition is like a station in QN, thus “the meeting point” of clients and servers. Assuming that there may be many or few of each one of them, fluidization can be considered for clients, for servers or for both. Table 1 represents the four theoretically possible cases. Finite server semantics corresponds at conceptual level to a *hybrid* behaviour: fluidization is applied only to clients, while servers are kept as discrete, counted as a finite number (the firing speed is bounded by the

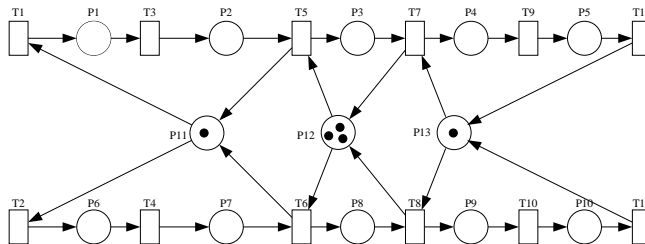


Fig. 2. A simplification of a production line in [20]

product of the speed of a server and the number of servers in the station). On the other hand, infinite server semantics really relaxes both clients and servers.

Table 1. The four cases for possible continuization of a transition [11]. The third one corresponds to *delays* in QN.

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 <i>implicit places</i> )
many (C)	many (C)	Infinite server semantics (speed is enabling-driven)

In both cases, although the fluidization is total, models are hybrid in the sense that they are piecewise linear systems, in which switching among the embedded linear systems is not externally driven as in [19], but internally through the minimum operators. If the PN belongs to the join-free subclass (i.e., transitions cannot have more than one input place), its fluidization generates a linear differential equation system.

The simple (asymmetric choice) model in Fig. 2 represents a production system with two production lines (it is part of an example studied in [20]). It has been analyzed using infinite server semantics and finite server semantics (single-server). In both cases, as the marking of the net is multiplied by a constant  $k$ , the throughput (normalized in the infinite server case) approaches to the one obtained when the net is seen as continuous (Table 2). In this case, the continuous net under infinite server semantics is a better approximation of the behaviour of the original discrete system. The finite server semantics disregards the restrictions due to the shared resources, that in this example are quite important unless the system is heavily loaded (large  $k$ ). Since the two semantics are not comparable, an immediate question is: Do they provide really different performance measures? The answer is positive, being possible to have large differences in the throughput (finite server semantics being usually more optimistic). Therefore, for a particular case, which continuous semantics is better? How much error can we have? For these questions we have no definitive answer, and it is not clear

**Table 2.** Steady state throughput of the system in Fig. 2, assuming each operation takes 1 t.u. (For  $k=10$  and  $k=50$ , markovian simulations are used.)

$k$	Reachable markings	Infinite servers		Single server
		Thr. of every $t_i$	Thr./ $k$	Thr. of every $t_i$
1	250	0.172	0.172	0.172
2	6300	0.366	0.183	0.303
3	67375	0.564	0.188	0.399
5	2159136	0.966	0.193	0.528
10	?	1.96	0.196	0.693
50	?	9.97	0.199	0.91
...	...	...	...	...
Continuous			0.2	1

if some in depth understanding can be obtained; nevertheless, an experimental analysis of benchmark examples from the literature is being considered [20]. As a preliminary remark, it seems that in most cases infinite server semantics provides a better approximation, although it seems difficult to obtain a characterization of the cases in which this happens. Moreover, other models —particularly useful for population dynamic problems— obtained through decoloration of colored models lead to a different semantics in which the “min” operator is replaced by a *product* [9] (which naturally keeps positive the decolored model).

### 3 Alternative Formalisms

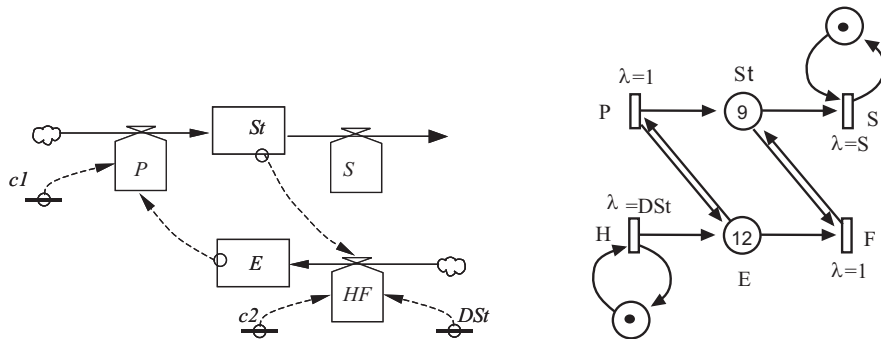
The approximation of a discrete event model by a continuous one is not new, and can be found in different formalisms.

Deterministic first-order approximations have been long used in QN [21, 3, 22, 4]. The fluid QN arises as a limit, in the sense of functional strong law of large numbers, of the stochastic network with the appropriate scaling. In [23] it was proven that the fluid models of certain QN could be used to analyze the (positive) recurrence of their discrete counterparts. In the last years, this has been extended to open or closed multiclass networks under different policies. Fluid models have also been applied in the synthesis of controls or scheduling (see, for example, [5, 6, 24–26]), proving that the optimal policy of the fluid networks can be somehow translated to a “good” policy for some discrete QN. In this setting, the emphasis has been usually put on the rigorous mathematical justification, even if that meant that fluid models were applied to “narrow” net classes.

Comparisons of PN and (monoclass) QN can be seen in [27–29]. Both formalisms are in essence *bipartite*: Places and transitions for PN; Queues and stations for QN. From an structural point of view, the main differences are the possible simultaneous existence in a single PN model of arc weights, attributions, choices, forks and joins, and the possible absence of local conservation rules when transitions are fired. QN are in essence timed models, that may be provided with very rich routing service and queuing disciplines, while PN can

be studied also as autonomous (idea of non-determinism). Moreover, as it was be pointed out in Subsect. 2.2, different timed interpretations of the net lead to different firing/flow policies. The firing logic of PN is of the type *consumption/production*, a kind of generalization of the classical *client/servers* in QN (Jackson, Gordon-Newell).

Forrester Diagrams (FD) appeared in the Systems Dynamics framework, and is a well-known, essentially continuous timed formalism for modelling certain classes of DES [30, 31]. FD provide a graphic representation of the system, that corresponds to an hydrodynamic interpretation. A comparison of PN and FD can be found in [32, 33]. For example, Fig. 3 represents both a FD and a PN model of an example. It is a production system that maintains a certain stock (St, with initial value 9) and a number of employees (E, with initial value 12). Products decrease due to sales, which are constant in time ( $S=12$ ), and increases with production (P), which is proportional to the number of employees (here with constant 1). Employees change due to hiring or firing (HF), which is proportional to the difference between the desired stock ( $DSt=10$ ) and the actual stock (St). Two remarks with respect to the PN model of the system: (1) variables have to be positive, hence variable HF has been split in two, hiring and firing; (2) to represent that sales are constant, or that firing depends on the stock, self-loops have been added with large enough arc weights to guarantee that these places always define the enabling degree (here weight 1 is enough). Although the continuous net is not structurally bounded, it is bounded with this timing. That is, complementary places ( $\bar{St}, \bar{E}$ ) with a large enough marking could be added without changing the behaviour. Continuous PN and FD provide a graphical support to



**Fig. 3.** A system described using FD and PN [33]. In the Forrester diagram, St and E are deposits with levels, while P, S and HF are valves regulating the flows.

generate systems of differential equations easily and there is a clear correspondence among their main types of nodes — place/level and transition/valve (or firing-speed/flow-variable). However, there are some differences [32, 33]:



- *Marking of places vs levels.* In FD each level corresponds to a state variable. Although in PN places are essentially state variables, redundancies may exist due to token conservation laws. Particular cases are structural implicit places. Levels may be negative, but markings are in essence positive.
- *Transitions vs. valves: flows.* The evolution of flows in FD takes place according to the information that valves receive from the whole state of the system, through the information channels. The evolution of the PN takes place according to the information that each transition receives from its input places. Thus, FD separate the *material* and the *information* flows, and evolve according to global information of the system. On the other hand, PN have only a flow of material that carries the information implicitly, and evolve according to information that, in standard uses, is local to each transition (its input places).
- In FD synchronizations are not explicitly modelled: there exist no elements to represent “rendez-vous”, and must be simulated by means of flow equations.
- In FD material is strictly conservative around the valves (the relationship among input and output flow is always 1:1), while in PN weighted conservation is often found.

As QN, FD are timed models. In FD the methodological analysis is basically focused to simulation, although there are some researching groups who also did go into the mathematical analysis of the system, basically sensibility, bifurcations and qualitative analysis (see for example [34]). Contrarily to QN, in FD the relationship between the solution of the continuous model and the original discrete model has not received much attention.

Under both, finite and infinite server semantics, “unforced” PN models are *positive systems* in Luenberger sense [35,36], that is, the non-negativity constraints on the marking are redundant. A particularly interesting case of positive systems are *compartmental systems*, which are composed of a finite number of subsystems (compartments), interacting by exchanging material among the compartments and with the environment [37–39].

An immediate similarity between PN and compartmental systems is that both allow representations based on *graphs*. However, PN are bipartite graphs, while compartmental models have a single kind of nodes. In discrete PN there are two different kinds of nodes: OR nodes (attributions/choices), and AND nodes (joins/forks). Nevertheless, in continuous PN (under infinite server semantics) the forward OR node in homothetic conflicts (if between  $t_1$  and  $t_2$ ,  $\mathbf{Pre}[P, t_1] = \alpha \cdot \mathbf{Pre}[P, t_2]$ ) is transformed into a “+” operation. Therefore, choices can be seen as *flow splitters*.

As in FD, in compartmental systems there is a strong “strict” conservation law: matter is not created, although it may “evaporate” and disappear if the system is not (output) closed. In PN such kind of constraint does not exist.

Another difference between PN and the graphs associated to compartmental systems is the arc weights. However, this is not a real generalization in the case of continuous nets *without synchronizations* (join-free nets), since for any of those nets, an equivalent one exists with arc weights one [36]. In other words, weights in

continuous models without synchronizations constitute a modelling convenience (i.e., do not add theoretical expressive power).

Putting all together, strongly connected and conservative PN without synchronizations are equivalent, from the modelling point of view, to closed linear compartmental systems [36].

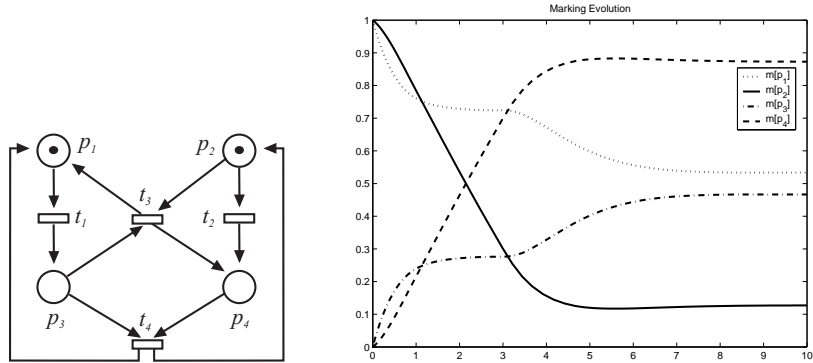
The above consideration of alternative “fluidified” formalisms is a source of opportunities in order to bridge the results in the analysis or synthesis of continuous PN models with that in “close” or related paradigms.

## 4 Performance Evaluation

Analyzing the performance of a continuous PN both in the transient and in the steady state involves integrating a set of differential equations. In theory it is possible to solve it analytically: solve the linear differential equations defined by the initial marking, and study among the different “minimums” associated to the synchronizations which one will be reached before, then repeat the process. In practice the existence of many differential equations, and many synchronizations makes this “artisanal” approach unfeasible, although a numerical version can be easily implemented in a computer, for example using Matlab. The equations that define the behaviour of the system in Fig. 4(a) are:

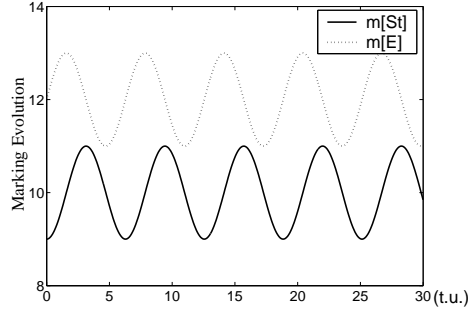
$$\begin{aligned} \mathbf{f}(\tau)[t_1] &= \boldsymbol{\lambda}[t_1] \cdot \mathbf{m}(\tau)[p_1] \\ \mathbf{f}(\tau)[t_2] &= \boldsymbol{\lambda}[t_2] \cdot \mathbf{m}(\tau)[p_2] \\ \mathbf{f}(\tau)[t_3] &= \boldsymbol{\lambda}[t_3] \cdot \min(\mathbf{m}(\tau)[p_2], \mathbf{m}(\tau)[p_3]) \\ \mathbf{f}(\tau)[t_4] &= \boldsymbol{\lambda}[t_4] \cdot \min(\mathbf{m}(\tau)[p_3], \mathbf{m}(\tau)[p_4]) \end{aligned}$$

If it is only the steady state we care about, some results and techniques have



**Fig. 4.** Continuous PN and the evolution of the marking with  $\boldsymbol{\lambda} = [0.50.110.3]$

been developed (see [10]). First of all, it has to be remarked that, in general, there is no guarantee about the existence of a steady state. For example, the net in Fig. 3(b) oscillates indefinitely without, even asymptotically, approaching to a steady state [33](see Fig. 5). To the best of our knowledge, the existence of a steady state had always be assumed, even for hybrid nets. In the future some work will be needed for the development of at least necessary or sufficient conditions. In the rest of this section some results for the computation of the steady state will be given, being assumed the existence of a steady state marking under infinite server semantics. Nevertheless, it should be pointed out that the markovian discrete counterpart of this model is ergodic. Let  $\mathbf{m}_{ss}$  be the steady



**Fig. 5.** The net system in Fig. 3(b) does not have a steady state.

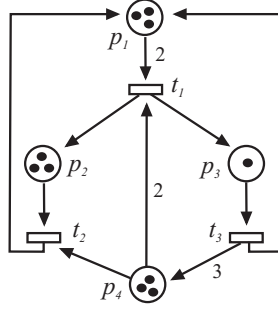
state marking of a bounded continuous net system:  $\mathbf{m}_{ss} = \lim_{\tau \rightarrow \infty} \mathbf{m}(\tau)$ . Then, for every  $\tau > 0$  it must be true:

$$\begin{aligned}
 \dot{\mathbf{m}}(\tau) &= \mathbf{C} \cdot \mathbf{f}(\tau) \\
 \mathbf{f}[t](\tau) &= \lambda[t] \cdot \min_{p \in \bullet t} \left\{ \frac{\mathbf{m}[p](\tau)}{\mathbf{Pre}[p, t]} \right\} \quad \forall \text{ non-immediate transition } t \\
 \mathbf{R} \cdot \mathbf{f}(\tau) &= \mathbf{0} \\
 \mathbf{m}(0) &= \mathbf{m}_0
 \end{aligned} \tag{1}$$

Using  $\phi$  as an approximation of  $\mathbf{f}$  in the steady state, and  $\mu$  as an approximation of the marking in the steady state, the above equations can be relaxed as follows:

$$\begin{aligned}
 \mu &= \mathbf{m}_0 + \mathbf{C} \cdot \sigma \\
 \phi[t] &= \lambda[t] \cdot \min_{p \in \bullet t} \left\{ \frac{\mu[p]}{\mathbf{Pre}[p, t]} \right\} \quad \forall \text{ non-immediate transition } t \\
 \mathbf{R} \cdot \phi &= \mathbf{0} \\
 \mathbf{C} \cdot \phi &= \mathbf{0} \\
 \mu, \sigma, \phi &\geq \mathbf{0}
 \end{aligned} \tag{2}$$

With this relaxation we have replaced the condition of being a reachable marking with that of being a solution of the fundamental equation. That is, we are losing the information about the feasibility of the transient path. Observe that the system is non-linear (“min” operator) and a unique solution is not guaranteed. For example, for the net system in Fig. 6 with  $\lambda = [2, 1, 1]$ , any marking  $[10 - 5 \cdot \alpha, 4 \cdot \alpha - 3, \alpha, \alpha]$ , with  $1 \leq \alpha \leq 5/3$ , verifies (2), and all of them lead to different throughput. Maximizing the flow of a transition, an *upper*



**Fig. 6.** A net system for which, with  $\lambda = [2, 1, 1]$ , infinite solutions of (2) exist.

*bound* of the throughput is obtained:

$$\begin{aligned} & \max\{\phi[t_1] \mid \mu = \mathbf{m}_0 + \mathbf{C} \cdot \sigma \\ & \phi[t] = \lambda[t] \cdot \min_{p \in \bullet t} \left\{ \frac{\mu[p]}{\mathbf{Pre}[p, t]} \right\} \forall \text{ non-immediate transition } t \\ & \mathbf{R} \cdot \phi = \mathbf{0} \\ & \mathbf{C} \cdot \phi = \mathbf{0}, \\ & \mu, \sigma, \phi \geq \mathbf{0} \} \end{aligned} \quad (3)$$

Notice that the solution of (3) is always dynamically “reachable” in the sense that with a suitable initial distribution of the tokens inside the P-semiflows, this throughput can be obtained (for instance, with the steady state distribution). Nevertheless, the programming problem in (3) is not easy to solve due to the “min” operator, that makes it *non linear*. The problem is that there is no way to know in advance which of the input places will restrict the flow, and so, a kind of branch and bound algorithm is used to solve it [10]. The idea is to solve the LPP defined by the system of (in)-equalities that appears choosing one input place per transition as the one which gives the minimum. 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 LPP that appear when each one of the input places are assumed to be defining the flow. That is, build a set of LPP 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. Some results have been developed that help to further prune the tree [10], but nevertheless, worst cases may be computationally expensive.

This suggests to go on with an additional relaxation, knowing that some accuracy may be lost. Since the minimum is the hardest point, that equation in (3) can be replaced with:

$$\phi_{ss}[t] = \lambda[t] \cdot \frac{\mu_{ss}[p]}{\mathbf{Pre}[p, t]} \quad \text{if } p = \bullet t \quad (4)$$

$$\phi_{ss}[t] \leq \lambda[t] \cdot \frac{\mu_{ss}[p]}{\mathbf{Pre}[p, t]} \quad \forall p \in \bullet t \quad \text{otherwise} \quad (5)$$

$$\frac{\phi_{ss}[t_i]}{\lambda[t_i]} = \frac{\phi_{ss}[t_j]}{\lambda[t_j]} \quad \forall t_i, t_j \text{ in EQ relation} \quad (6)$$

This way we have a single linear programming problem (LPP), thus computation is of polynomial time complexity. Unfortunately, the LPP provides in general a non tight bound, i.e, the solution may be non reachable for any distribution of the tokens verifying the P-semiflow load conditions,  $\mathbf{y} \cdot \mathbf{m}_0$ . This may occur because it may be the case that for that solution none of the input places of a synchronization really restricts the flow of that transition. When this happens, the marking cannot define the steady state (the flow of that transition would be larger). See [10] for conditions that guarantee in some systems the reachability of the bound.

A slight relaxation of that LPP, using inequalities in all the transitions, leads to a result that had been obtained long before for discrete nets. For bounded *discrete* net systems, an upper bound of the throughput of one transition can be obtained by means of that LPP [40, 41].

$$\begin{aligned} \max\{\phi[t_i] \mid & \mu = \mathbf{m}_0 + \mathbf{C} \cdot \sigma \\ & \phi[t] \leq \lambda[t] \cdot \frac{\mu[p]}{\mathbf{Pre}[p, t]} \quad \forall p \in \bullet t \\ & \mathbf{R} \cdot \phi = \mathbf{0} \\ & \mathbf{C} \cdot \phi = \mathbf{0}, \\ & \mu, \sigma, \phi \geq \mathbf{0}\} \end{aligned} \quad (7)$$

For nets in which the steady state visit ratio can be deduced from the structure and the rates of the transitions (mono T-semiflow reducible nets), it can

be proved that both statements are equivalent (in the sense that if one solution with inequality exists, another one with the same throughput verifies the equality) [10].

## 5 Parametric Design

In engineering, analysis techniques frequently guide in the definition of synthesis methods. Assuming approximate computation of performance with efficient algorithms, the problem of designing the best set of resources (best  $\mathbf{m}_0$ ), the best routing policy (best  $\mathbf{R}$ ), the best type of machines (machine selection problem, appearing indirectly as determining the best  $\lambda$ ) can be straightforwardly stated. Observe that these are not “on-line” control problems, but “off-line” design problems in which parametric optimizations are being solved. A recent presentation of some “easily” (i.e., polynomial time) solvable problems of this type can be found in [11]. One of the basic statements is as follows: Let  $\mathbf{g} \cdot \mathbf{f} - \mathbf{w} \cdot \mathbf{m} - \mathbf{b} \cdot \mathbf{m}_0$ , be the profit function to be optimized, where,  $\mathbf{g}$  represents a gain vector associated to flows/throughput,  $\mathbf{w}$  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, ...). Assume also upper bounds in the use of the resources ( $\mathbf{V} \cdot \mathbf{m}_0 \leq \mathbf{k}$ ), i.e., its availability is limited.

This kind of optimization problem admits a particularly elegant and efficient solution if the LPP, stated in Sect. 4 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 [9]). 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,  $\mathbf{R}$* . The following LPP can be written:

$$\begin{aligned}
& \max \{ \mathbf{g} \cdot \phi - \mathbf{w} \cdot \mu - \mathbf{b} \cdot \mu_0 \\
& \text{s.t. } \mu = \mu_0 + \mathbf{C} \cdot \sigma \\
& \quad \phi[t] \leq \lambda[t] \cdot \frac{\mu[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in \bullet t \\
& \quad \phi[t] = \lambda[t] \cdot \frac{\mu[p]}{\mathbf{Pre}[p,t]} \quad \text{if } p = \bullet t \\
& \quad \mathbf{C} \cdot \phi = \mathbf{0} \\
& \quad \mathbf{R} \cdot \phi = \mathbf{0} \\
& \quad \phi, \mu, \mu_0 \geq \mathbf{0} \\
& \quad \mathbf{V} \cdot \mu_0 \leq \mathbf{k} \}
\end{aligned} \tag{8}$$

In other cases the problem to be solved is which are the minimum cost resources ( $\mathbf{b} \cdot \mathbf{m}_0$ ) that guarantee a certain throughput (see Problem 5 in [11]). The routing matrix  $\mathbf{R}$  may be also the parameters to be optimized, looking for best production mix, or better internal routing at the factory. A simple case for optimizing a profit function w.r.t. the routing  $\mathbf{R}$  is the following example:

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,  $\boldsymbol{\phi}$ , being  $\mathbf{R}$  free.

$$\begin{aligned}
& \max\{ \mathbf{g} \cdot \boldsymbol{\phi} - \mathbf{w} \cdot \boldsymbol{\mu} - \mathbf{b} \cdot \boldsymbol{\mu}_0 \\
& \text{s.t. } \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \\
& \quad \phi[t] \leq \lambda[t] \cdot \frac{\mu[p]}{\text{Pre}[p,t]} \quad \forall p \in \bullet t \\
& \quad \phi[t] = \lambda[t] \cdot \frac{\mu[p]}{\text{Pre}[p,t]} \quad \text{if } p = \bullet t \\
& \quad \mathbf{C} \cdot \boldsymbol{\phi} = \mathbf{0} \\
& \quad \boldsymbol{\mu} \geq \mathbf{0} \}
\end{aligned} \tag{9}$$

Once LPP (9) has been solved, the computation of the routing matrix  $\mathbf{R}$ , is straightforward, just proceed free-choice by free-choice. Assuming for simplicity that choices are binary:  $\phi_1/\phi_2 = r_1/r_2$ , and  $r_1 + r_2 = 1$ .

If all free conflicts are solved with immediate transitions, and  $\mathbf{g} = \mathbf{1}$ ,  $\mathbf{w} = \mathbf{b} = \mathbf{0}$ , this LPP is analogous to the one stated in [42], assuming boundedness. Even if in this last case nets are P-timed (i.e., with delays associated to places), and conflicts are solved according to a stationary routing policy (a simplifying preselection policy, which in practice is equivalent a net without conflicts), and have different transient behaviour, their steady state is the same.

## 6 Observation

In order to control a dynamic system, frequently it is necessary to know its current state. Sensors can be used to get information from the plant, but often some of the variables cannot be directly measured, either because it is not physically possible, or because of its cost. If the information that can be obtained from the system allows to estimate the value of a variable, that variable is said to be observable (with that instrumentation), and the estimate constitutes the observation. The *observability problem*, i.e., the characterization of which state variables are observable and its observation, has been studied both for continuous systems (in particular linear systems) and for discrete event systems. Some results related to observability of discrete event models can be found in [43, 44].

With respect to continuous systems, observability is quite a classical problem, for which easy to understand and general results were obtained for time-invariant linear systems in the sixties of the last century [45, 35]. The contribution of inputs to the evolution of a linear system can be easily computed and subtracted from the total output. Therefore, observability of linear systems can be studied using its unforced counterpart.

A time-invariant linear system can be expressed as  $\dot{\mathbf{x}}(\tau) = \mathbf{A} \cdot \mathbf{x}(\tau) + \mathbf{B} \cdot \mathbf{u}(\tau)$ ,  $\mathbf{y}(\tau) = \mathbf{S} \cdot \mathbf{x}(\tau)$ , where  $\mathbf{y}$  represents its output, that is, “what is seen” of it. A linear system is said to be *observable* iff knowing  $\mathbf{y}(\tau)$ , it is possible to compute its initial state  $\mathbf{x}(\tau_0)$ . That is, iff knowing  $\mathbf{y}(\tau)$  the equation

$$\mathbf{y}(\tau) = \mathbf{S} \cdot e^{\mathbf{A} \cdot \tau} \cdot \mathbf{x}(\tau_0)$$

can be solved for every  $\mathbf{x}(\tau_0)$ . It can be seen that this is equivalent to matrix  $\vartheta = (\mathbf{S}^T | (\mathbf{S}\mathbf{A})^T | \dots | (\mathbf{S}\mathbf{A}^{n-1})^T)^T$  having full rank. This result is known as the

*observability theorem* and the matrix is known as the *observability matrix* [45]. For linear systems, the observable subspace can be characterized algebraically. Intuitively, a system state estimation can be *theoretically* obtained from the output signal and the computation of its derivatives.

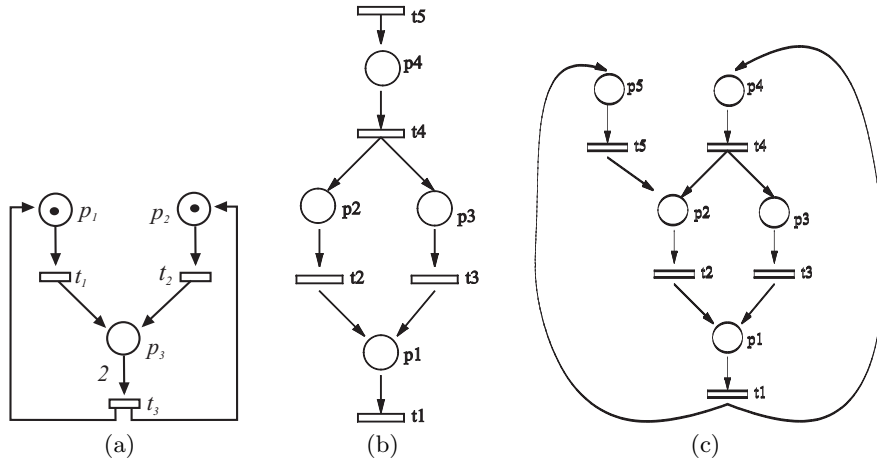
Hence observability is completely characterized for nets that can be described with a linear system, that is, join-free nets (i.e., nets that do not have synchronizations). If the net has synchronizations, there are several linear systems that may define the evolution of the system, depending on which is the place that restricts each transition. The observability theorem has been extended to general piecewise linear systems. The complete system is observable iff the pairwise intersection of different observable subspaces is trivial, that is, the joint observability matrix of each pair of linear system has full rank [46]. However, continuous PN have a characteristic that these general piecewise systems do not have: the change of one linear system to another one is triggered by the continuous state (the marking). This makes the observability of continuous PN a more simple issue [47]: if the system passes through an observable linear system, its marking at that moment can be observed. And since it is deterministic, it is possible to simulate it backwards and deduce the initial marking.

Notice also that observability of a synchronization will not be possible in general unless *all its input places* are measured (it might be possible to measure one place only if it were timed implicit). Moreover, observability cannot be extended forward (the output flow of a transition does not provide information to deduce the marking of the next place). Hence, the problem can be tackled by measuring the places in synchronizations. The net that remains removing those places and their input and output arcs—eventually composed of several unconnected subnets—is join-free. For these subnets, the observability theorem can be applied. Hence, given a set of measured places it is not difficult to prove whether the net is observable or not.

Nets without *synchronizations and attributions* ( $p \in P$  is an attribution if  $|\bullet p| > 1$ ) can be observed just measuring the “final” places (places without output arcs) or measuring one (any) place if it is a weighted cycle. As a direct consequence, it can be stated that a weighted T-system is observable for any initial marking iff all synchronization places are measured, or, in the case of a cycle, one arbitrary place is measured [48]. For this kind of nets, the rates of the transitions do not have any influence on the observability of the system. *Attributions* on the other hand, force to study the rates of the net. Observe for example the net in Fig. 7(a), and assume that  $p_3$  is measured. The system is observable iff the rates of  $t_1$  and  $t_2$  are not equal, i.e.  $\lambda[t_1] \neq \lambda[t_2]$ . Intuitively, if they have the same rate, it is not possible to distinguish which part of the flow is coming from each place [47]. Moreover, this is not a local problem, but a global one. On the one hand, it is not just the rates of the input transitions of the attribution that have to be taken into account. For example, the net in Fig. 7(b) is not observable if  $\lambda[t_2] = \lambda[t_3]$ , but neither is if  $\lambda[t_4] = \frac{2 \cdot \lambda[t_2] \cdot \lambda[t_3]}{\lambda[t_2] + \lambda[t_3]}$ . On the other hand, attributions are not “independent”. For example, the net in Fig. 7(c) with  $\lambda[t_2] \neq \lambda[t_3]$  and  $\lambda[t_4] = \lambda[t_5] = \frac{2 \cdot \lambda[t_2] \cdot \lambda[t_3]}{\lambda[t_2] + \lambda[t_3]}$  is observable if  $p_4$  is



measured, but not if  $p_5$  is measured. For any other value of  $\lambda[t_4] = \lambda[t_5]$  it is observable measuring either  $p_4$  or  $p_5$ . A related “design” problem is to determine



**Fig. 7.** Observability in nets with attributions depends on both the rates, and the structure

minimal cost observability. That is, when a cost is assigned to measuring each place, which is the best selection of the places guaranteeing that the system is observable? To apply the previous result would mean to solve a combinatorial set of observability problems. Nevertheless this number can be greatly reduced in many cases applying the following property [48]: Let  $p$  and  $p'$  be such that there is a path from  $p'$  to  $p$  without synchronizations or attributions. Then

- $p'$  can be deduced from the observation of  $p$ .
- If the net is not observable measuring a set of places containing place  $p$ , it cannot be observable if  $p$  is replaced by  $p'$ .

An algorithm, and its application to an example showing how the combinatorics is reduced can be found in [48].

## 7 Dynamic Control: on “forced” continuous net systems

In order 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 something new; the same strategy is used for QN, 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 non 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  $\alpha \cdot \lambda[t]$ , with  $\alpha \in (0, 1)$ ) or *additive* (subtracting  $\mathbf{u}$ ,  $\mathbf{0} \leq \mathbf{u} \leq \mathbf{f}$ ). In any case, the flow can go from  $\mathbf{f}[t]$  to  $\mathbf{0}$ . That is, the control can locally slow down the activity of transitions. 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 fundamental equation is now:  $\dot{\mathbf{m}} = \mathbf{C} \cdot (\mathbf{f}(\mathbf{m}) - \mathbf{u})$ , 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 (and restrictive) sense of [35, 37] (see [36]); (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 [49]. For the present purpose let us just point out that if all transitions are controllable, reachability in timed models is equivalent to reachability in the underlying untimed models [50]. 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 controlled timed model.

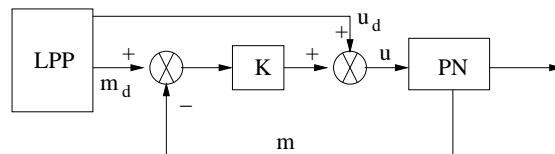
Notice that it will take infinite time to reach a marking that empties a place, unless perhaps if it were already empty in the initial marking. However, that kind of markings will always have at least one transition with 0 throughput, hence they are not very interesting as steady state markings. Once again, this problem is not something new: the loading of a capacity in a basic RC-electrical circuit cannot be “complete” in finite time. Nevertheless, engineers use the classical concept of response time (at 5%, 3%, 1%) in order to have a practical view of the duration of the transient behaviour.

Let us assume in the sequel that *all* the transitions are controllable, and let us concentrate first in the steady state control. A first remark is that given a net and a constant steady state control,  $\mathbf{u}$ , there may exist several markings, perhaps with different flows, that may be steady state markings. For example, for the system depicted in Fig. 6, with  $\lambda = [2, 1, 1]$ , and  $\mathbf{u} = \mathbf{0}$ , marking  $[5, 1, 1, 1]$  (with flow  $[1, 1, 1]$ ) and marking  $[5/3, 11/3, 5/3, 5/3]$ , (with flow  $[5/3, 5/3, 5/3]$ ) can be both steady states. Hence, a first interesting step is to obtain the optimum steady state, and a control action for it. A LPP similar to the one in (9) can be used for that [50]:

$$\begin{aligned}
& \max\{ \mathbf{g} \cdot \phi - \mathbf{w} \cdot \boldsymbol{\mu} - \mathbf{b} \cdot \boldsymbol{\mu}_0 \\
& \text{s.t. } \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \\
& \quad \phi[t] = \lambda[t] \cdot \frac{\boldsymbol{\mu}[p]}{\mathbf{Pre}[p, t]} - \mathbf{v}[p, t] \quad \forall p \in \bullet t \\
& \quad \mathbf{C} \cdot \phi = \mathbf{0} \\
& \quad \boldsymbol{\mu}, \boldsymbol{\sigma} \geq \mathbf{0} \\
& \quad \mathbf{v} \geq \mathbf{0} \}
\end{aligned} \tag{10}$$

The only difference is that now a set of slack variables  $\mathbf{v}[p, t]$  have been added in the equation that relates the marking with the flow. If  $|\bullet t| = 1$ , the slack variable  $\mathbf{v}[\bullet t, t]$  represents the control action of the transition. In general, it can be seen that  $\mathbf{u}[t] = \min_{p \in \bullet t} \mathbf{v}[p, t]$  is an appropriate control input.

With respect to the transient, the use of  $\mathbf{u}$  as the single reference input is not enough in general to reach the optimal flow in the net. In [11, 50] a schema as the one in Fig. 8 is proposed, in which the control action depends on the steady state control action, and the difference between the actual marking and the steady state one.



**Fig. 8.** Control schema

One approach that has been used in the literature to face the problem of optimal control of hybrid systems is to approximate them by discrete-time systems, and represent them as Mixed Logical Dynamical (MLD) systems [51]. Usually, in a MLD system the time step is constant. Time discretization has two important drawbacks: (1) The length of the sampling period is not easy to define. There exists a tradeoff between accuracy (short sampling period) and computational speed (long sampling period). In fact, the complexity typically grows exponentially with the number of switching variables, and these, for a given time interval, are inversely proportional to the length of the sampling period; (2) It is assumed that events can occur only at time instants that are multiple of the sampling period. In fact, it would be desirable to deal with a model that requires a minimum number of steps (samples) without losing accuracy.

In [52] it was seen that the behaviour of finite server semantics continuous PN system could be described by means of an MLD system. Moreover, since PN are event-driven systems, it could be a continuous-time event-driven MLD, instead of one that evolves with a fixed time step. Observe that this approach has two interesting advantages: (1) Event-discretization does not imply loss of accuracy: The marking evolution of a continuous PN is linear between events, and so it can be determined from the marking of the net at the event instants. (2) The number of steps is minimized: A step happens only when it is really required (an event happens).

Different kinds of optimal control problems can be solved by means of the explained event-driven approach, for example: reaching a target marking in minimum time, i.e., time optimal control, maximizing the steady state throughput, or maximizing an optimization function in which several different parameters are involved [52].

Some preliminary efforts are also being made to transform the optimal control problem into a multi-parametric quadratic program, and apply the techniques developed for this kind of systems [19].

## Acknowledgments

We would like to thank the following people for their contributions to this research line: E. Teruel, J. Júlvez, E. Jiménez, C. Mahulea and D. Rodríguez.

## References

1. Silva, M., Colom, J.M.: On the computation of structural synchronic invariants in P/T nets. In Rozenberg, G., ed.: *Advances in Petri Nets 1988*. Volume 340 of *Lecture Notes in Computer Science*. Springer (1988) 387–417
2. Silva, M., Teruel, E., Colom, J.M.: Linear algebraic and linear programming techniques for the analysis of net systems. In Rozenberg, G., Reisig, W., eds.: *Lectures in Petri Nets. I: Basic Models*. Volume 1491 of *Lecture Notes in Computer Science*. Springer (1998) 309–373
3. Kleinrock, L.: *Queuing Systems, Volume II: Computer Applications*. Volume 2. Wiley (1976)
4. Mandelbau, A., Chen, H.: Discrete flow networks: Bottleneck analysis and fluid approximations. *Mathematical Operations Research* **16** (1991) 408–446
5. Chen, H., Yao, D.: *Fundamentals of Queueing Networks. Performance, Asymptotics and Optimization*. Volume 46 of *Applications of Mathematics. Stochastic Modelling and Applied Probability*. Springer (2001)
6. Cassandras, C.G., Sun, G., Panayiotou, C.G., Wardi, Y.: Perturbation analysis of multiclass stochastic fluid models. In: *15<sup>th</sup> IFAC World Congress, Barcelona, Spain* (2002)
7. David, R., Alla, H.: Continuous Petri nets. In: *Proc. of the 8th European Workshop on Application and Theory of Petri Nets, Zaragoza, Spain* (1987) 275–294
8. David, R., Alla, H.: *Discrete, Continuous, and Hybrid Petri Nets*. Springer-Verlag (2004)
9. Silva, M., Recalde, L.: Petri nets and integrality relaxations: A view of continuous Petri nets. *IEEE Trans. on Systems, Man, and Cybernetics* **32** (2002) 314–327
10. Júlvez, J., Recalde, L., Silva, M.: Steady-state performance evaluation of continuous mono-t-semiflow Petri nets. *Automatica* **41** (2005) 605–616
11. Silva, M., Recalde, L.: On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control* **28** (2004) 253–266
12. Nakamura, M., Silva, M.: An iterative linear relaxation and tabu search approach to minimum initial marking problems of timed marked graphs. In: *Procs. of European Control Conference, ECC99, Aachen, Germany* (1999)
13. Silva, M.: Introducing Petri nets. In: *Practice of Petri Nets in Manufacturing*. Chapman & Hall (1993) 1–62
14. Recalde, L., Teruel, E., Silva, M.: Autonomous continuous P/T systems. In Donatelli, S., Kleijn, J., eds.: *Application and Theory of Petri Nets 1999*. Volume 1639 of *Lecture Notes in Computer Science*, Springer (1999) 107–126
15. Silva, M.: Towards a synchrony theory for P/T nets. In Voss, K., et al., eds.: *Concurrency and Nets*. Springer (1987) 435–460

16. Molloy, M.K.: Performance analysis using stochastic Petri nets. *IEEE Trans. on Computers* **31** (1982) 913–917
17. Recalde, L., Silva, M.: Petri Nets fluidification revisited: Semantics and steady state. *European Journal of Automation APII-JESA* **35** (2001) 435–449
18. Alla, H., David, R.: Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers* **8** (1998) 159–188
19. Bemporad, A., Giua, A., Seatzu, C.: An iterative algorithm for the optimal control of continuous-time switched linear systems. In Silva, M., Giua, A., Colom, J., eds.: *WODES 2002: 6th Workshop on Discrete Event Systems*, Zaragoza, Spain, IEEE Computer Society (2002) 335–340
20. Mahulea, C., Rodríguez, D., Recalde, L., Silva, M.: Choosing server semantics for continuous Petri nets: An examples driven approach. Research report, Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna, 1, 50018 Zaragoza, Spain (2005) Forthcoming.
21. Newell, G.F.: *Applications of Queueing Theory*, Second Edition. Chapman and Hall (1982)
22. Vandergraft, J.: A fluid model of networks of queues. *Management Science* **29** (1983) 1198–1208
23. Dai, J.: On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability* **5** (1995) 49–77
24. Weiss, G.: Scheduling and control of manufacturing systems — a fluid approach. In: *Proceedings of the 37 Allerton Conference*. (1999) 577–586
25. Meyn, S.P.: Sequencing and routing in multiclass queueing networks. part I: Feedback regulation. *SIAM Journal on Control and Optimization* **40** (2002) 741–776
26. Meyn, S.P.: Sequencing and routing in multiclass queueing networks part II: Workload relaxations. *SIAM Journal on Control and Optimization* **42** (2003) 178–217
27. Vernon, M., Zahorjan, J., Lazowska, E.D.: A comparison of performance Petri nets and queueing network models. In: *3rd Int. Workshop on Modelling Techniques and Performance Evaluation*, Paris, France (1987)
28. Silva, M., Campos, J.: 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. (1993) 14–21
29. Chiola, G.: Petri nets versus queueing networks. [53] chapter 4 121–134
30. Forrester, J.W.: *Industrial Dynamics*. MIT Press, Cambridge, Mass. (1961)
31. Forrester, J.W.: *Urban Dynamics*. Productivity Press (1969)
32. Jiménez, E., Recalde, L., Silva, M.: 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)*. (2001) 85–94
33. Jiménez, E., Júlvez, J., Recalde, L., Silva, M.: Relaxed continuous views of discrete event systems: considerations in Forrester diagrams and Petri nets. In: *Proc. of the Int. Conf. on Systems, Man and Cybernetics (SMC 2004)*, The Hague, The Netherlands (2004)
34. Mosekilde, E., Aracil, J., Allen, P.: Instabilities and chaos in nonlinear dynamic systems. *Systems Dynamics Review* **4** (1988) 14–55
35. Luenberger, D.G.: *Introductions to Dynamic Systems: Theory, Models and Applications*. John Wiley and Sons, New York (1979)
36. Silva, M., Recalde, L.: Unforced continuous Petri nets and positive systems. In Benvenuti, L., Santis, A.D., Farina, L., eds.: *Positive Systems. Proceedings of the First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA 2003)*. Volume 294 of LNCIS., Rome, Italy, Springer (2003) 55–62

37. Farina, L., Rinaldi, S.: Positive Linear Systems. Theory and Applications. Pure and Applied Mathematics. John Wiley and Sons, New York (2000)
38. Benevenuti, L., Farina, L.: Positive and compartmental systems. *IEEE Transactions on Automatic Control* **47** (2002) 370–373
39. Walter, G., Contreras, M.: Compartmental Modeling With Networks. Birkhauser Boston (1999)
40. Chiola, G., Anglano, C., Campos, J., Colom, J., Silva, M.: Operational analysis of timed Petri nets and application to the computation of performance bounds. In: *Proc. of the 5<sup>th</sup> Int. Workshop on Petri Nets and Performance Models (PNPM93)*, Toulouse, France, IEEE Computer Society Press (1993) 128–137
41. Campos, J.: Performance bounds. [53] chapter 17 587–636
42. Gaujal, B., Giua, A.: Optimal routing of continuous timed Petri nets. In: *15<sup>th</sup> IFAC World Congress*, Barcelona, Spain (2002)
43. Ramirez-Trevino, A., Rivera-Angel, I., Lopez-Mellado, E.: Observability of discrete event systems modeled by interpreted Petri nets. *IEEE Trans. on Robotics and Automation* **19** (2003) 557–565
44. Giua, A., Seatzu, C.: Observability of Place/Transition nets. *IEEE Trans. on Automatic Control* **47** (2002) 1424–1437
45. Luenberger, D.G.: Observers for multivariable systems. *IEEE Trans. on Automatic Control* **AC-11** (1966) 190–197
46. Vidal, R., Chiuso, A., Soatto, S., Sastry, S.: Observability of linear hybrid systems. In: *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003*, April 3-5, 2003, Proceedings. Volume 2623 of Lecture Notes in Computer Science., Prague, Czech Republic, Springer (2003) 526–539
47. Júlvez, J., Jiménez, E., Recalde, L., Silva, M.: On observability in timed continuous Petri net systems. In G. Franceschinis, J.K., Woodside, M., eds.: *Proc. of the 1st Int. Conf. on the Quantitative Evaluation of Systems (QEST 2004)*, Enschede, The Netherlands, IEEE Computer Society Press (2004) 60–69
48. Mahulea, C., Recalde, L., Silva, M.: Optimal observability for continuous Petri nets. In: *16<sup>th</sup> IFAC World Congress*, Prague, Czech Republic (2005) To appear.
49. Jiménez, E., Júlvez, J., Recalde, L., Silva, M.: On controllability of timed continuous petri net systems: the join free case. Research report, Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna, 1, 50018 Zaragoza, Spain (2005) Submitted to CDC-ECC’05.
50. Mahulea, C., Ramirez-Trevino, A., Recalde, L., Silva, M.: On the control of continuous Petri nets under infinite server semantics. Research report, Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna, 1, 50018 Zaragoza, Spain (2005)
51. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. *Automatica* **35** (1999) 407–427
52. Júlvez, J., Bemporad, A., Recalde, L., Silva, M.: Event-driven optimal control of continuous Petri nets. In: *43rd IEEE Conference on Decision and Control (CDC 2004)*, Paradise Island, Bahamas (2004)
53. Balbo, G., Silva, M., eds.: *Proc. of Human Capital and Mobility MATCH—Performance Advanced School*. In Balbo, G., Silva, M., eds.: *Performance Models for Discrete Event Systems with Synchronizations: Formalisms and Analysis Techniques*, Jaca, Spain (1998)