Chapter 16
Introduction to fluid Petri nets

C. Renato Vázquez, Cristian Mahulea, Jorge Júlvez, and Manuel Silva

Abstract The fluidization of Petri nets is a relaxation technique introduced in the literature in order to avoid the so called *state explosion problem*. This technique results in a model, named continuous or fluid, whose state variables (the marking at the places) take real nonnegative values. Among the implications, the fluidization opens the possibility for transferring concepts and techniques from the continuous-state systems paradigm to the discrete event systems one. In this chapter, the concept of non-forced continuous Petri net is recalled, together with the most usual firing semantics for the timed interpretation of this model. Certain properties on the resulting fluid models are described, remarking limitations and advantages found in the analysis of fluid Petri nets. The focus is on the relationships between the discrete and the continuous Petri nets. It is also discussed the quality of the relaxation, from qualitative and quantitative perspectives. In particular, questions like the preservation of liveness, boundedness or the marking evolution, are addressed.

16.1 Introduction and motivation

Several results can be found in the literature regarding the analysis of DEDS by using models from the Petri net (PN) paradigm. Applications involve the implementation of sequence controllers, validation in software development, analysis of communication protocols and networks, manufacturing systems, supply chains, etc.

It is well known that one of the most important limitations in the analysis (and synthesis problems) of DEDS is the computational complexity that may appear. In particular, the set of reachable markings of a Petri net frequently increases exponentially w.r.t. the initial marking, what is known as the *state explosion problem*, making prohibitive the application of enumerative techniques even for net systems with a small structure (i.e., small number of places and transitions).

The authors are with the Instituto de Investigación en Ingeniería de Aragón (I3A), University of Zaragoza, Spain, e-mail: {cvazquez,cmahulea,julvez,silva}@unizar.es

In this context, the *fluidization* or *continuization* (i.e., getting a continuous-state approximation) has been proposed as a relaxation technique in order to overcome the state explosion problem. The idea consists in the analysis of the DEDS via a relaxed continuous approximation, i.e., a continuous-state system if behaves in a "similar" way than the original model (or conserves certain interesting properties), reducing thus the computational efforts. Nevertheless, not all PNs models allow some continuous approximations. In DEDS, fluidization has been explored in queueing networks (e.g., [5], [8], [1]), PNs ([9], [30]) and more recently in Process Algebra [15].

Regarding PNs, David and Alla firstly introduced fluid PNs with *constant* and *variable speeds* [9, 10]. From another perspective, the relaxation of the fundamental or state equation of the PN system was proposed in [29] (in the same meeting), in order to systematically use Linear Programming for structural analysis. These two approaches lead to continuous state equations (see Fig. 16.1), but the proposal (more conceptual) of David and Alla leads to the possibility of describing the transient behavior of timed models. The resulting fluid PN models can be analyzed as state-continuous systems but behave (quantitatively) as T-timed discrete PNs (examples can be found in [11]). This topic was revisited in [24], making emphasis in the connection with the original discrete models. In fact, there the *infinite server semantics* (which is the same that the variable speed) was derived as the approximation of the average behavior of a Markovian stochastic T-timed PN (a PN whose transitions fire after exponentially distributed random time delays). From another perspective, different authors have proposed hybrid PN systems (some transitions remain discrete while the others are fluidified) that can be used as models *per se*, for instance *fluid stochastic PNs* [35], *differential PNs* [4], *batch hybrid PNs* [12], *first-order hybrid PNs* [2], etc. These hybrid models enjoy a broad representative power, but the analysis of some of these systems is technically complex.

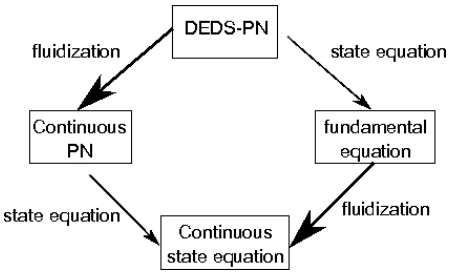

Fig. 16.1 Two ways for the fluidization of Petri nets.

In this and the following chapters, the basic model in [11, 24] will be mostly considered in both its autonomous version (untimed) and timed version (mostly under infinite server semantics or variable speed). Then, from our perspective, a continuous Petri net model is derived as a potential approximation of an original discrete

PN. In this way, the analysis of these continuous models provides coherence, synergy and economy with respect to the Petri net paradigm [32, 33].

The fluidization seems a promising technique when the initial marking can be assumed as "large enough"(where the relative errors, and their consequences, tend to be small, because the rounding effects are relatively less significant). In fact, increasing the population does not affect the complexity of the analysis via fluid models, since, in the resulting continuous PN, the number of *state variables* is upper bounded by the number of places, being independent of the number of tokens in the net system (thus, of the initial marking).

The risk of fluidization is a loss of fidelity. As linearization of non-linear models, fluidization does not always lead to relaxed models of similar behavior, i.e., they are not always approximations. The first problem that may arise when using this approach is that the fluid model does not necessarily preserve all the behavioral properties of the original DEDS model. For example, mutex properties (e.g., two places cannot be concurrently marked) are always lost, spurious markings (solutions of the state equation that are not reachable markings in the discrete PN) may appear in the continuous model or liveness is not preserved in the general case. Thus, for certain cases, the analysis through fluidization may be useless. In other cases, the fluidization may provide only an educated guess. Moreover, the resulting fluid model may exhibit an important technical complexity. For instance, a timed continuous PN under infinite server semantics is a *piecewise linear* system (a linear system whose state and input matrices change, among a countable set of matrices, according to the state), from a continuous-state dynamical systems' perspective. The number of embedded linear operation modes (equivalently, sets of linear differential equations) usually increases exponentially w.r.t. the number of transitions representing rendez-vous (synchronizations). Thus, even if the behavior of a DEDS is approximately preserved by its corresponding fluid relaxation, this could be still too complex to be properly analyzed. The number of state variables does not depend on the initial marking, but in the number of places. Thus, large net structures may lead to continuous models with a large number of state variables, since one state variable is usually defined per each place. Furthermore, the addition of time in the continuous model brings important additional difficulties for analysis and synthesis. In this sense, the expressive power of timed continuous PNs (under infinite server semantics) is surprisingly high, because they can simulate Turing Machines [23]! This means that certain important properties as the existence of a steady-state are undecidable.

On the other hand, when a system admits a "reasonable" fluidization (in the sense that the fluid model preserves the desired properties of the discrete one), several advantages can be visualized by using continuous models: the first one is obviously the reduction of the complexity related to a large marking population, since in continuous models the state explosion problem does not appear. Furthermore, certain problems can be analyzed by using more efficient algorithms, for instance, the set of reachable markings (including infinite firing sequences) is convex, thus reducing the complexity of optimization problems. Additionally, the ability to fire in isola-

tion minimal T-semiflows makes behavioral and structural synchronic relations [1] to collapse, thus, for example, the bound of a place can be straightforwardly computed in polynomial time [28, 29]. Another interesting advantage is that techniques and concepts developed in the Control Theory for continuous-sate systems can be applied to the timed continuous PN model. For instance, techniques for the analysis and application of performance controllers that reject perturbations, stability, observability, estimation, etc. In this way, fluidization represents a bridge between particular classes of continuous-state and discrete event systems.

In this chapter, *continuous Petri nets* are firstly introduced as untimed, i.e., fully non-deterministic, and later as timed formalisms. The relationship between the properties of (discrete) PNs and the corresponding properties of their continuous approximation is considered at several points. Observability and Control of *timed continuous Petri nets* (TCPNs) will be considered in forthcoming chapters.

16.2 Fluidification of Untimed net models

This section presents the formalism of continuous Petri nets and its behavior in the untimed framework. It deals with basic concepts, as lim-reachability [27] and desired logical properties, and relates them to those ones of the discrete systems.

16.2.1 The continuous PN model

A continuous Petri net system [11, 25] is understood as the fluid relaxation of all the transitions of a *discrete* Petri net one (as a consequence, the marking at all the places becomes continuous). In the sequel, the set of input and output nodes of v will be denoted as ${}^\bullet v$ and $v^\bullet$, respectively.

Definition 1 *A* continuous PN *system is a pair* $\langle \mathcal{N}, m_0 \rangle$ *where* $\mathcal{N}$ *is a P/T net (like in a P/T system) and* $m_0 \in \mathbb{R}_{\geq 0}^{|P|}$ *is the initial marking. The evolution rule is different to the case of discrete P/T systems, since in continuous PNs the firing is not restricted to integer amounts, and so the marking* $m \in \mathbb{R}_{\geq 0}^{|P|}$ *is not forced to be integer. Instead, a transition* t_i *is* enabled *at* m *iff for every* $p_j \in {}^\bullet t_i$, $m[p_i] > 0$*; and its* enabling degree *is* $enab(t_i, m) = \min_{p_j \in {}^\bullet t_i} \{m[p_j]/Pre[p_j, t_i]\}$. *The firing of* t_i *in a certain amount* $0 \leq \alpha \leq enab(t_i, m)$ *leads to a new marking* $m' = m + \alpha \cdot C[P, t_i]$, *where* $C = Post - Pre$ *is the* token flow *or* incidence *matrix, and* $C[P, t_i]$ *denotes the column of* C *corresponding to* t_i.

[1] Synchrony theory is a branch of general net theory that deals with the characterization of transition firing dependencies. Two transition subsets are in a given synchronic relation if the corresponding quantitative firing dependency is bounded. Behavioral relations depend on the initial marking, while structural relations hold for any (finite) initial marking.

The usual PN system, $\langle \mathscr{N}, M_0 \rangle$ with $M_0 \in \mathbb{N}^{|P|}$, will be said to be *discrete* so as to distinguish it from a *continuous* PN system $\langle \mathscr{N}, m_0 \rangle$, in which $m_0 \in \mathbb{R}_{\geq 0}^{|P|}$. In the following, the marking of a continuous PN will be denoted in lower case m, while the marking of the corresponding *discrete* one will be denoted in upper case M. Observe that $Enab(t_i, M) \in \mathbb{N}$ in discrete PNs, while $enab(t_i, m) \in \mathbb{R}_{\geq 0}$ in continuous PNs. Notice that to decide whether a transition in a continuous system is enabled or not it is not necessary to consider the weights of the arcs going from the input places to the transition. However, the arc weights are important to compute the enabling degrees. Here no policy for the firing of transitions is imposed, that is, a full non-determinism is assumed for the order and amounts in which transitions are fired.

Right and left natural annullers of the token flow matrix are called T- and P-*semiflows*, respectively (i.e., vectors y and x, whose entries belong to $\mathbb{N} \cap \{0\}$, fulfilling $y^T \cdot C = 0$ and $C \cdot x = 0$, respectively). The existence of P-semiflows induces conservation laws, i.e., if $\exists y \geq 0, y^T \cdot C = 0$ then by the state equation it holds $y^T \cdot m_0 = y^T \cdot m$ for any initial marking m_0 and any reachable marking m. On the other hand, T-semiflows represent potential cyclic behaviours, i.e., if $\exists x \geq 0, C \cdot x = 0$ then $\exists m_0$ such that $m_0 \xrightarrow{\sigma} m_0$ with σ being a firing sequence whose firing count vector equals x. As in discrete nets, when $y^T \cdot C = 0$ for some $y > 0$ the net is said to be *conservative*, and when $C \cdot x = 0$ for some $x > 0$ the net is said to be *consistent*. Given a vector $x \in \mathbb{R}^{|T|}$, its support is defined as the set of transitions $||x|| = \{t_i \in T | x[i] \neq 0\}$, where $x[i]$ denotes the ith entry of x. Similarly, for a vector $y \in \mathbb{R}^{|P|}$, $||y|| = \{p_i \in P | y[i] \neq 0\}$.

The definitions of subclasses that depend only on the structure of the net are also generalized to continuous nets. For instance, in *join free* nets (JF) each transition has at most one input place, in *choice free* nets (CF) each place has at most one output transition, and in *equal conflict* nets (EQ) all conflicts are equal, i.e., ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow Pre[P,t] = Pre[P,t']$. Moreover, a net $\mathscr{N}$ is said to be *proportional equal conflict* if ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \exists q \in \mathbb{R}_{>0}$ such that $Pre[P,t] = q \cdot Pre[P,t']$. Finally, a net $\mathscr{N}$ is said to be *mono-T-semiflow* (MTS) iff it is conservative and has a unique minimal T-semiflow whose support contains all the transitions (a T-semiflow is minimal if its support does not contain the support of another T-semiflow).

16.2.2 Reachability

Let us now illustrate the firing rule in an untimed continuous Petri net system. For this, consider the system in Fig. 16.2(a). The only enabled transition at the initial marking is t_1, whose enabling degree is 1. Hence, it can be fired in any real quantity going from 0 to 1. For example, by firing it an amount equal to 1, the marking $m_1 = [1\ 1]^T$ is reached. At m_1 transition t_2 has enabling degree equal to 1; if it is fired in an amount of 0.5, the resulting marking is $m_2 = [1.5\ 0.5]^T$. In this way, both m_1 and m_2 are reachable markings with *finite* firing sequences. On the other hand, at m_1 the marking at p_1 is equal to 1, leading to an enabling degree for t_1 of 0.5,

i.e., the half amount that at m_0. By firing t_1 an amount of 0.5, the marking reached is $m_3 = [0.5\ 1.5]^T$. Notice that by successively firing t_1 with an amount equal to its enabling degree, the marking of p_1 will approach to 0. The marking reached in the limit (with an infinite firing sequence), namely $m' = [0\ 2]^T$, corresponds to the emptying of an initially marked trap $\Theta = \{p_1\}$ (a trap is a set of places $\Theta \subseteq P$ s.t. $\Theta^\bullet \subseteq {}^\bullet\Theta$), fact that can not occur in discrete systems. Thus, in continuous systems traps may not *trap*! Nevertheless, such trap cannot be emptied with a finite firing sequence. This leads us to consider two different reachability concepts.

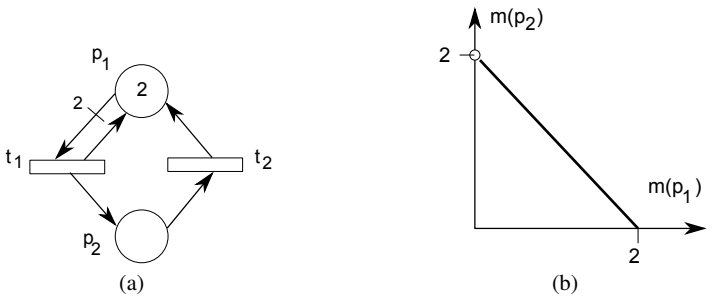

Fig. 16.2 (a) Autonomous continuous system (b) Lim-Reachability space

Definition 2 *A marking is said to be* reachable *if it is reachable with a* finite *firing sequence. On the other hand, a marking is said to be* lim-reachable *if it can be reached with either a* finite *or an* infinite *firing sequence. More formally:* $m \in \mathbb{R}_{\geq 0}^{|P|}$ *is* lim-reachable *if a sequence of reachable markings* $\{m_i\}_{i\geq 1}$ *exists such that* $m_0 \xrightarrow{\sigma_1} m_1 \xrightarrow{\sigma_2} m_2 \cdots \cdots m_{i-1} \xrightarrow{\sigma_i} m_i \cdots$ *and* $\lim_{i\to\infty} m_i = m$. *For a given system* $\langle \mathcal{N}, m_0 \rangle$, *the set of all markings that are reachable in a finite number of firings is denoted as* $\mathrm{RS}(\mathcal{N}, m_0)$, *while* $\text{lim-RS}(\mathcal{N}, m_0)$ *denotes the set of* lim-*reachable markings.*

As an example, Fig. 16.2(b) depicts $\text{lim-RS}(\mathcal{N}, m_0)$ of the system in Fig. 16.2(a). The set of lim-reachable markings is composed of the points on the line connecting $[2\ 0]^T$ and $[0\ 2]^T$. On the other hand, all the points of that line excepting the circled one $[0\ 2]^T$ belong to $\mathrm{RS}(\mathcal{N}, m_0)$, i.e., $\mathrm{RS}(\mathcal{N}, m_0) = \text{lim-RS}(\mathcal{N}, m_0) \setminus \{[0\ 2]^T\}$.

For any continuous system $\langle \mathcal{N}, m_0 \rangle$, the differences between $\mathrm{RS}(\mathcal{N}, m_0)$ and $\text{lim-RS}(\mathcal{N}, m_0)$ are just in the border points on their convex spaces. In fact, it holds that $\mathrm{RS}(\mathcal{N}, m_0) \subseteq \text{lim-RS}(\mathcal{N}, m_0)$ and that the closure of $\mathrm{RS}(\mathcal{N}, m_0)$ is equal to the closure of $\text{lim-RS}(\mathcal{N}, m_0)$ [16].

16.2.3 Some advantages

A system can be fluidizable with respect to a given property, i.e., the continuous model preserves that property of the discrete one, but may be not with respect to

other properties. Thus, the usefulness of continuous models highly depends on the properties to be analyzed and the systems being studied.

An interesting property is that $\mathrm{RS}_D(\mathscr{N},m_0) \subseteq \mathrm{RS}(\mathscr{N},m_0)$ (where $\mathrm{RS}_D(\mathscr{N},m_0)$ is the set of markings reachable by the system as discrete). This can be explained as follows: for any marking m reached by firing transitions in discrete amounts from $m_0 \in \mathbb{N}^{|P|}_{\geq 0}$, i.e., as if the system were discrete, m is also reachable by the system as continuous just by applying the same firing sequence.

The fact that $\mathrm{RS}_D(\mathscr{N},m_0) \subseteq \mathrm{RS}(\mathscr{N},m_0)$ might involve a mismatch among some properties of the discrete and continuous systems, e.g., the new reachable markings might make the system live or might deadlock it (examples can be found in [27, 31]). This *non-fluidizability* of discrete net systems with respect to the deadlock-freeness property, that may be surprising at first glance, can be easily accepted if one thinks, for example, on the existence of non-linearizable differential equations systems .

Let us recall the concept of boundedness in discrete Petri nets: a *PN* system is said *bounded* if $k \in \mathbb{N}$ exists such that for every reachable marking m, $m \leq k \cdot 1$, with 1 is the vector of ones, and it is *structurally bounded* if it is bounded with every initial marking. These concepts can also be applied to continuous Petri net systems. Similarly, a continuous system is said lim-live if for any transition t and any lim-reachable marking m, a successor m' exists such that t is enabled [27]. A continuous net $\mathscr{N}$ is said structural lim-live if there exists an initial marking m_0 such that the continuous system $\langle \mathscr{N},m_0\rangle$ is lim-live.

The concept of limit-reachability in continuous systems provides an interesting approximation to liveness properties of discrete nets, in the sense that lim-liveness and boundedness of the continuous system is a sufficient condition for structural liveness and structurally boundedness of the discrete one [27]:

Proposition 3 *Let $\langle \mathscr{N},m_0\rangle$ be a bounded lim-live continuous PN system. Then, $\mathscr{N}$ is structurally live and structurally bounded as a discrete net.*

From Proposition 3 it is clear that any necessary condition for a discrete system to be str. live and str. bounded, is also necessary for it to be str. lim-live and bounded as continuous. In particular rank theorems [26] establish necessary liveness conditions based on consistency, conservativeness and the existence of an upper bound on the rank of the token flow matrix, which is the number of equal conflict sets. Even more, for the EQ subclass, such structural conditions are also sufficient for lim-liveness [27].

There are some interesting advantages when dealing with fluid PN models. An important one is that the reachability set $\mathrm{RS}(\mathscr{N},m_0)$ is *convex* [27].

Proposition 4 *Let $\langle \mathscr{N},m_0\rangle$ be a continuous PN system. The set $\mathrm{RS}(\mathscr{N},m_0)$ is convex, i.e., if two markings m_1 and m_2 are reachable, then for any $\alpha \in [0,1]$, $\alpha m_1 + (1-\alpha)m_2$ is also a reachable marking.*

Notice that in a continuous system any enabled transition can be fired in a sufficiently small quantity such that it does not become disabled. This implies that every transition is fireable if and only if a strictly positive marking is reachable. This is

equivalent to the non existence of empty *siphons* (a siphon is a set of places $\sum$ s.t. ${}^\bullet\sum \subseteq \sum^\bullet$, then an unmarked siphon cannot gain marks). From this, realizability of T-semiflows can be deduced [27], and therefore behavioral and structural synchronic relations [28] coincide in continuous systems in which every transition is fireable at least once. In particular, considering boundedness and structural boundedness as in discrete systems, both concepts coincide in continuous systems in which every transition is fireable. And, as in discrete systems, structurally boundedness is equivalent to the existence of $y > 0$ such that $y \cdot C \leq 0$ (see, for example, [3, 34]).

Another interesting property is that $\mathrm{RS}_D(\mathscr{N}, m_0) \subseteq \mathrm{RS}(\mathscr{N}, m_0)$ implies that boundedness of the continuous system is a sufficient condition for boundedness of the discrete one. Moreover, it is important to stress that the set lim-$\mathrm{RS}(\mathscr{N}, m_0)$ can be easily characterized if some common conditions are fulfilled [27].

Proposition 5 *Let $\langle \mathscr{N}, m_0 \rangle$ be consistent and such that each transition can be fired at least once. Then $m \in$ lim-$\mathrm{RS}(\mathscr{N}, m_0)$ iff there exists $\sigma > 0$ such that $m = m_0 + C \cdot \sigma$.*

Hence, if a net is consistent and all the transitions are fireable, then the set of lim-reachable markings is fully characterized by the state equation. This immediately implies convexity of lim-$\mathrm{RS}(\mathscr{N}, m_0)$ and the inclusion of every spurious discrete solution in lim-$\mathrm{RS}(\mathscr{N}, m_0)$. Fortunately, every spurious solution in the border of the convex set lim-$\mathrm{RS}(\mathscr{N}, m_0)$ can be *cut* by adding some implicit places (more precisely the so-called *cutting implicit places* [7]) what implies clear improvements in the state equation representation. This will be detailed in Subsection 16.4.1.

On the other hand, if $\langle \mathscr{N}, m_0 \rangle$ is not consistent or some transitions cannot be fired, lim-$\mathrm{RS}(\mathscr{N}, m_0)$ can still be characterized by using the state equation plus a simple additional constraint concerning the fireability of the transitions in $\|\sigma\|$. The set $\mathrm{RS}(\mathscr{N}, m_0)$ can also be fully determined by adding one further constraint related to the fact that a finite firing sequence cannot empty a trap [16] (in contrast to infinite sequences which might empty initially marked traps as shown in the previous section).

16.3 Fluidization of timed net models

This section introduces the notion of time in the continuous Petri net formalism presenting the most used firing semantics. The main focus will be on *infinite server semantics*. Some basic properties will be discussed.

16.3.1 Server semantics

If a timed interpretation is included in the *continuous* model (time is associated to the transitions, thus they fire with certain speed), the fundamental equation depends

on time: $m(\tau) = m_0 + C \cdot \sigma(\tau)$, which, by assuming that $\sigma(\tau)$ is differentiable, leads to $\dot{m}(\tau) = C \cdot \dot{\sigma}(\tau)$. The derivative of the firing count vector of the sequence σ is $f(\tau) = \dot{\sigma}(\tau)$, called the *(firing) flow*, and leads to the following equation for the dynamics of the timed continuous PN (TCPN) system:

$$\dot{m}(\tau) = C \cdot f(\tau). \tag{16.1}$$

Depending on how the flow f is defined, different firing *semantics* can be obtained. For *finite server semantics* (FSS), if the markings of the input places of t_j are strictly greater than zero (*strongly enabled*), its flow will be constant, equal to λ_j, i.e., all servers work at full speed. Otherwise (*weakly enabled*), the flow will be the minimum between its maximal firing speed and the total input flow to the input empty places. Formally,

$$f_j(\tau) = \begin{cases} \lambda_j & \text{if } \forall p_i \in {}^\bullet t_j, m[p_i] > 0 \\ min \left\{ \underset{p_i \in {}^\bullet t_j | m_i = 0}{min} \left\{ \sum_{t' \in {}^\bullet p_i} \frac{f[t'] \cdot Post[t', p_i]}{Pre[p_i, t_j]} \right\}, \lambda_j \right\} & \text{otherwise} \end{cases} \tag{16.2}$$

The dynamical system under FSS corresponds to a piecewise constant system and a switch occurs when a marked place becomes empty and the new flow values are computed ensuring that the marking of all places remain positive. Many examples using this semantics are given in [11] while in [20] a model is studied using this and the following semantics.

In the case of *infinite server semantics* (ISS), the flow of transition t_j is given by:

$$f_j(\tau) = \lambda_j \cdot enab(t_j, m(\tau)) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \frac{m_i}{Pre[p_i, t_j]}, \tag{16.3}$$

Under ISS, the flow through a transition is proportional to the marking of the input place determining the enabling degree. As already advanced, TCPNs under ISS have the capability to simulate Turing machines [23], thus they enjoy an important expressive power; nevertheless, certain important properties are *undecidable* (for example, marking coverability, submarking reachability or the existence of a steady-state).

The flow through a transition under *product* (population) semantics (PS) [2] is given by

$$f_j(\tau) = \lambda_j \cdot \prod_{p_i \in {}^\bullet t_j} \frac{m_i}{Pre[p_i, t_j]}, \tag{16.4}$$

Product semantics can lead to *chaotic* models, i.e., models of deterministic dynamical systems that are extremely sensitive to initial conditions. This semantic is frequently adopted in population systems, like predator/pray, biochemistry, etc. (see, for example, [14]).

[2] Through discoloration of colored (discrete) nets, the *minimum* operator of ISS is transformed into a PS [30].

In the case of manufacturing or logistic systems, it is natural to assume that the transition firing flow is the minimum between the number of clients and servers and, FSS or ISS are mainly used [31, 11]. Since, these two semantics provide two different approximations of the discrete net system, an immediate problem is to decide which semantics will approximate "better" the original system. In [11], the authors observed that most frequently ISS approximates better the marking of the discrete net system. Furthermore, for mono-T-semiflow reducible net systems [17], it has been proved that ISS approximates better the flow in steady state under some (general) conditions [20]. For this reason, ISS will be mainly considered in the rest of this work. Let us recall the formal definition.

Definition 6 *A* timed continuous Petri net *(TCPN) is a time-driven continuous-state system described by the tuple* $\langle \mathcal{N}, \lambda, m_0 \rangle$*, where* $\langle \mathcal{N}, m_0 \rangle$ *is a* continuous *PN and the vector* $\lambda \in \mathbb{R}_{>0}^{|T|}$ *represents the transitions rates that determine the temporal evolution of the system. Under* ISS *the* flow *(the firing speed) through a transition* t_i *is defined as the product of the rate* λ_i *and* $enab(t_i, m)$*, i.e.,* $f_i(m) = \lambda_i \cdot enab(t_i, m) = \lambda_i \cdot \min_{p \in {}^\bullet t_i} \{m[p]/Pre[p, t_i]\}$.

Let us recall now some useful concepts.

Definition 7 *A* configuration *of a TCPN, denoted as* $\mathscr{C}_k$*, is a set of* (p,t) *arcs, one per transition,* covering *the set* T *of transitions. It is said that the system at marking* m *is at a configuration* $\mathscr{C}_k$ *if the arcs in* $\mathscr{C}_k$ *provide the minimum ration in the ISS definition (16.3), or equivalently,* $\mathscr{C}_k$ *is the active configuration at* m*. A configuration matrix* $|T| \times |P|$ *is associated to each configuration as follows:*

$$\Pi_k[t_j, p_i] = \begin{cases} \frac{1}{Pre[p_i, t_j]}, & \text{if } (p_i, t_j) \in \mathscr{C}_k \\ 0, & \text{otherwise} \end{cases} \tag{16.5}$$

The reachability set of a TCPN system can be partitioned (except on the borders) according to the configurations, and inside each obtained convex region $\mathscr{R}_k$ *the system dynamics is linear. More formally,* $\mathscr{R}_k = \{m \in \text{lim-RS}(\mathcal{N}, m_0) | \Pi_k m \leq \Pi_j m, \forall \Pi_j \in \{\Pi\}\}$*, where* $\{\Pi\}$ *denotes the set of all configuration matrices.*

The number of regions and configurations is upper bounded by $\prod_{t \in |T|} |{}^\bullet t|$ but some of them can be redundant, thus removed [21]. Let us define the firing rate matrix $\Lambda = diag(\lambda)$ (i.e., a diagonal $|T| \times |T|$ matrix containing the rates of the transitions). The evolution of a TCPN, under ISS, can be represented as:

$$\dot{m}(\tau) = C \cdot f(\tau) = C \cdot \Lambda \cdot \Pi(m) \cdot m(\tau), \tag{16.6}$$

where $\Pi(m)$ is the configuration matrix operator ($\Pi(m) = \Pi_k$ where $\mathscr{C}_k$ is the active configuration at m). Notice that, while the system is evolving inside a region $\mathscr{R}_k$, it behaves linearly as $\dot{m} = C\Lambda\Pi_k m$, thus a TCPN under ISS is a piecewise-linear system.

16.3.2 Qualitative properties under ISS

According to (16.3), it is obvious to remark that being the initial marking of a continuous net system positive, the marking will remain positive during the (unforced or non-controlled) evolution. Hence, it is not necessary to add constraints to ensure the non-negativity of the markings. On the other hand, according to (16.3) as well, two homothetic properties are dynamically satisfied:

- if λ is multiplied by a constant $k > 0$ then identical markings will be reached, but the system will evolve k times faster;
- if the initial marking is multiplied by k, the reachable markings are multiplied by k and the flow will also be k times bigger.

Unfortunately, ISS has not only "good" properties and some "paradoxes" appear. For example, it could be thought that, since fluidization relax some restrictions, the throughput (flow at steady-state) of the continuous system should be at least that of the timed discrete one. However, the throughput of a TCPN is not in general an upper bound of the throughput of the discrete PN [31]. Moreover, if only some components of λ or only some components of m_0 are increased the steady state throughput is not monotone in general.

Two monotonicity results of the steady-state throughput are satisfied under some general conditions [20]:

Proposition 8 *Assume $\langle \mathcal{N}, \lambda_i, m_i \rangle$, $i = 1,2$ are mono-T-semiflow TCPNs under ISS which reach a steady-state. Assume that the set of places belonging to the arcs of the steady state configuration contains the support of a P-semiflow. If*

1. *$\langle \mathcal{N}, \lambda_1, m_1 \rangle$ and $\langle \mathcal{N}, \lambda_1, m_2 \rangle$ verify $m_1 \leq m_2$ or*
2. *$\langle \mathcal{N}, \lambda_1, m_1 \rangle$ and $\langle \mathcal{N}, \lambda_2, m_1 \rangle$ verify $\lambda_1 \leq \lambda_2$,*

then the steady state flows satisfy $f_1 \leq f_2$.

Let us consider for instance the mono-T-semiflow TCPN in Fig. 16.3(a) under ISS with $\lambda_1 = \lambda_3 = 1$ and $m_0 = [15\ 1\ 1\ 0]^T$. Different modes can govern the evolution of the system at steady-state. For example, if $0 < \lambda_2 \leq 0.5$, the flow in steady-state is $f_1(\tau) = m_1(\tau)$, $f_2(\tau) = m_4(\tau)$ and $f_3(\tau) = m_3(\tau)$, respectively (i.e., $\frac{m_1(\tau)}{2} < \frac{m_4(\tau)}{2}$ and $m_4(\tau) < m_2(\tau)$ in steady state). Therefore, $\mathscr{C}_2 = \{(p_1,t_1),(p_4,t_2),(p_3,t_3)\}$ is the steady-state configuration and the set of places $\{p_1,p_4,p_3\}$ determines (constrains) the flow. Since this set contains the support of the P-semiflow $y = [1\,0\,4\,1]$, the steady-state flow is monotone (Fig. 16.3(b)). When $\lambda_2 > 0.5$, the steady-state configuration becomes $\mathscr{C}_3 = \{(p_4,t_1),(p_2,t_2),(p_3,t_3)\}$, i.e., the set of places governing the evolution becomes $\{p_4,p_2,p_3\}$, that is the support of the P-flow $y = [0\,1\,-3\,-1]$, not a P-semiflow, and monotonicity may not hold (Fig. 16.3(b)).

The connection between liveness in the autonomous (untimed) continuous model and the timed ones has been investigated. First of all, notice that if a steady-state exists in the timed model, from (16.1) $\dot{m} = C \cdot f_{ss} = 0$ is obtained (independently of the firing semantics), where f_{ss} is the flow vector of the timed system in the steady

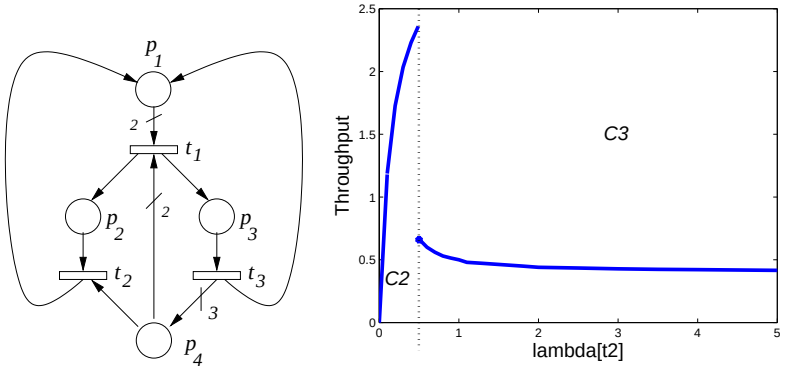

Fig. 16.3 A Mono-T-semiflow net and its "fluid" throughout in steady-state. Observe that it is not smooth, and that increasing $\lambda_2 > 0.5$ the throughput is counterintuitive (faster machine, slower behavior

state, $f_{ss} = \lim_{\tau\to\infty} f(\tau)$. Therefore, the flow in the steady state is a T-semiflow of the net. Deadlock-freeness and liveness definitions can be extended to timed continuous systems as follows:

Definition 9 *Let $\langle \mathcal{N}, \lambda, m_0 \rangle$ be a timed continuous PN system and f_{ss} be the vector of flows of the transitions in the steady state.*

- $\langle \mathcal{N}, \lambda, m_0 \rangle$ *is* timed-deadlock-free *if* $f_{ss} \neq 0$*;*
- $\langle \mathcal{N}, \lambda, m_0 \rangle$ *is* timed-live *if* $f_{ss} > 0$*;*
- $\langle \mathcal{N}, \lambda \rangle$ *is* structurally timed-live *if* $\exists\, m_0$ *such that* $f_{ss} > 0$.

Notice that if a timed system is not timed-live (timed-deadlock-free), it can be concluded that, seen as untimed, the system is not lim-live (lim-deadlock-free) since the evolution of the timed system just gives a particular trajectory that can be fired in the untimed system. This fact allows us to establish a one-way bridge from liveness conditions of timed and untimed systems. The reverse is not true, i.e., the untimed system can deadlock, but a given λ can *drive* the marking along a trajectory without deadlocks. In other words, the addition of an arbitrary transition-timed semantics to a system imposes constraints on its evolution what might cause the timed system to satisfy some properties, as boundedness and liveness, that are not necessarily satisfied by the untimed system [38, 37]. The relationships among liveness definitions are depicted in Fig. 16.4.

As an example, let us show how some conditions initially obtained for timed systems can be applied to untimed ones. It is known that if a MTS timed net $\langle \mathcal{N}, \lambda \rangle$ is structurally live for any $\lambda > 0$ then for every transition t there exists $p \in {}^\bullet t$ such that $p^\bullet = \{t\}$, i.e., p is *structurally persistent* or *conflict-free* [18]. Let $\langle \mathcal{N}, \lambda \rangle$ be a MTS timed net containing a transition t such that for every $p \in {}^\bullet t$, $|p^\bullet| > 1$. According to

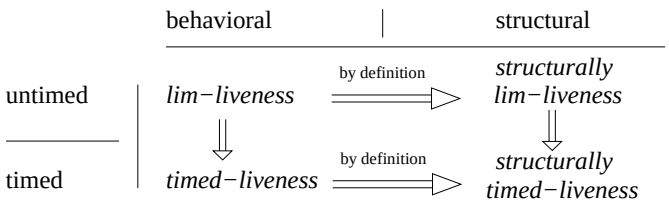

Fig. 16.4 Relationships among liveness definitions for continuous models

the mentioned condition λ exists such that $\langle \mathscr{N}, \lambda \rangle$ is not structurally timed-live. Therefore $\mathscr{N}$ is not structurally lim-live, since structurally timed-liveness is a necessary condition for structurally lim-liveness (see Fig. 16.4).

16.3.3 On the quantitative approximation under ISS

Fluid PNs are usually considered as relaxations of original discrete models. In fact, the definitions for the most usual semantics for timed continuous PNs were inspired by the average behavior of high populated timed discrete PNs [11, 25]. Nevertheless, the dynamic behavior of a timed continuous PN model does not always approximate that of the corresponding timed discrete PN. For this reason, it is important to investigate the conditions that lead to a valid relaxation. In some sense, this subsection deals with the *legitimization* of the so called ISS and the consideration of some issues that affect the quality of the approximation.

Let us consider Markovian Petri nets (MPN), i.e., stochastic discrete Petri nets with exponential delays associated to the transitions and conflicts solved by a race policy [22]. In [36] it was shown that, in certain cases, the marking of a TCPN under ISS approximates the average marking of the corresponding MPN (having the same structure, rates and intial marking, under the assumption of ergodicity). The approximation is better when the enabling degrees of the transitions (the number of active servers) is large and the system mainly evolves inside one marking region (according to one configuration or linear mode), i.e., for each synchronization, a single place is *almost always* constraining the throughput. Errors in the approximation may appear due to the existence of *sychronizations*: *arc weights* greater than one and *joins* (*rendez-vous*). The reason is that the enabling degree (thus the flow) definition for the TCPN does not accurately describe the enabling degree (thus the throughput) of the discrete model in these cases. In fact, the approximation is perfect for ordinary Join-Free Petri nets.

Let us provide an intuitive explanation about how the arc weights introduce approximation errors. Assume that the continuous marking of a TCPN approximates the average marking of the corresponding MPN at certain time τ, i.e., $E\{M(\tau)\} \sim m(\tau)$. Given an arc with weight q connecting a place p_j to a transition t_i, the *expected* enabling degree of t_i in the MPN would be $E\{Enab(t_i)\} = E\{\lfloor M[p_j]/q \rfloor\}$, which is different from the enabling degree in the TCPN $enab(t_i) = m[p_j]/q \sim E\{M[p_j]\}/q$,

due to the presence of the operator $\lfloor \cdot \rfloor$ (note that in ordinary arcs $q = 1$, thus $\lfloor M[p_j]/q \rfloor = M[p_j]/q$), and the quality of the approximation will be reduced or lost for future time.

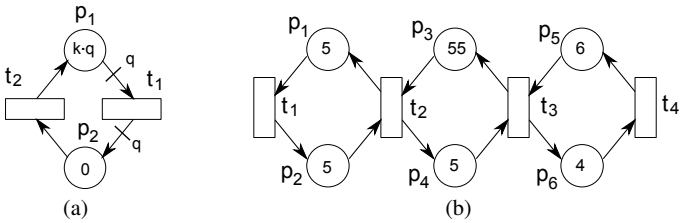

Fig. 16.5 a) Cycle net with arc weights. b) Marked graph which evolves through different regions.

(a)

$k = 1 \setminus q =$	1	2	4	8	16
$MPN, \chi[t_1]$	0.50	0.40	0.32	0.26	0.22
$TCPN, f[t_1]$	0.50	0.50	0.50	0.50	0.50

(b)

$q = 4 \setminus k =$	1	2	4	8	16
$MPN, \chi[t_1]$	0.32	0.80	1.76	3.78	7.68
$TCPN, f[t_1]$	0.50	1.00	2.00	4.00	8.00

Table 16.1 Throughput and its approximation for t_1 of the net of Fig. 16.5(a).

As an example, consider the MPN system of Fig. 16.5(a) with timing rates $\lambda_1 = \lambda_2 = 1$, and initial marking $M_0 = [k \cdot q,\ 0]^T$, where $k, q \in \mathbb{N}^+$. This system, and its corresponding TCPN, were evaluated for different values of k and q. The obtained values for the throughput and flow of t_1, at steady state, are shown in table 16.1. Note that, when $k = 1$ (table 16.1(a), here the marking is relatively very small), the larger the weight of the input arc of t_1 (i.e., q), the larger the difference (error) between the throughput in the MPN ($\chi[t_1]$) and the flow in the TCPN ($f[t_1]$). Observe that the flow in the continuous model remains unchanged. On the other hand, when the arc weights are fixed but the initial marking (i.e., k) is increased (table 16.1(b)), the relative approximation error decreases (in such case, $E\{\lfloor M[p_1]/q \rfloor\} \sim E\{M[p_1]\}/q$ for $M[p_1] >> q$). Concluding, the relative errors introduced by arc weights become smaller when the marking in the net is increased w.r.t. those weights.

Let us illustrate now how joins (rendez-vous) introduce approximation errors. Given a synchronization t_i with two input places $\{p_j, p_k\}$, the expected enabling in the MPN would be $E\{Enab(t_i)\} = E\{min(M[p_j], M[p_k])\}$, which is not equal to the enabling in the TCPN $enab(m[p_j], m[p_k]) \sim min(E\{M[p_j]\},\ E\{M[p_k]\})$, because the order in which the expect value and the "min" operator are applied cannot be commuted.

As an example, the MPN system of Fig. 16.5(b) was simulated with timing rates $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and different rates for t_4: $\lambda_4 \in \{2, 1.5, 1.2, 1\}$. The corresponding TCPN model was also simulated. The average markings at the steady state are shown in table 16.2 (columns MPN and TCPN). The column denoted as

$E\{Enab(t_4)\}$ is the average enabling degree of t_4 at the steady state (this represents a lower bound for the number of active servers in the transitions). The value in column $P(M \in \mathscr{R}_{ss})$ is the probability that the marking is inside the region $\mathscr{R}_{ss}$, related to the steady state of the TCPN (equivalently, the fraction of time that $M(\tau)$ evolves according to a single configuration or linear mode). Note that the lower the probability that $M(\tau)$ belongs to $\mathscr{R}_{ss}$, the larger the difference (the error) between the MPN and the TCPN, even if the average enabling degrees increase. On the other hand, a good approximation is provided when the probability that $M(\tau) \in \mathscr{R}_{ss}$ is high, which occurs for $\lambda_4 = 2$. The approximation holds because, in this case, $M(\tau)$ mainly evolves in one region $\mathscr{R}_{ss}$ (in particular, $E\{min(M[p_4], M[p_5])\} \sim E\{M[p_4]\}$ and $E\{min(M[p_2], M[p_3])\} \sim E\{M[p_2]\}$).

Table 16.2 Marking approximation of p_3 for the MPN of Fig. 16.5(b).

λ_4	MPN	TCPN	TnCPN	$E\{Enab(t_4)\}$	$P(M \in \mathscr{R}_{ss})$
2	54.62	55	54.63	2.53	0.8433
1.5	53.87	55	53.88	3.22	0.661
1.2	51.16	55	51.17	3.88	0.413
1	29.97	55	30.73	4.93	0.036

From a continuous-systems perspective, it can be deduced that approximation does not accumulate in time if the steady state marking of the continuous model is *asymptotically stable* (because the deviations of the MPN from its expected behavior, which is similar to that of the TCPN, vanish with the time evolution). Therefore, asymptotic stability is a necessary condition (together with *liveness*, otherwise, the continuous system may die while the discrete is live) for the approximation of the steady state.

16.4 Improving the approximation: removing spurious solutions, addition of noise

Since the approximation provided until now by a fluid PN is not always accurate, a question that may arise is the possibility of improving such approximation by means of modifying the continuous Petri net definition. Through this section, a couple of approaches, for such improvement, will be discussed.

16.4.1 Removing spurious solutions

The state equation provides a full characterization of the lim-reachable markings (in the autonomous continuous model) for consistent nets with no empty siphons. This

allows one to use the state equation to look for deadlocks, i.e., markings at which every transition has at least one empty input place. In some cases, at such a deadlock m there is an empty trap that was initially marked. Nevertheless, it is well known that initially marked traps cannot be completely emptied in discrete nets. Thus, m is a *spurious solution* of the state equation if we consider the system as discrete, equivalently, deadlock-freeness has not been preserved during fluidization.

Consider for instance the net in Fig. 16.6 with $m_0 = [10\ 11\ 0]^T$. The marking $m = [0\ 1\ 10]^T$ is a deadlock and can be obtained as a solution of the state equation with $\sigma = [10\ 0]^T$ as firing count vector. Thus given that the system satisfies the conditions of Proposition 5, m is lim-reachable, i.e., the system lim-deadlocks. Notice that p_1 is a trap (${}^\bullet p_1 = p_1{}^\bullet$) that was initially marked and can be emptied by an infinite firing sequence. Thus, m is not reachable in the discrete net, thus it is an spurious deadlock.

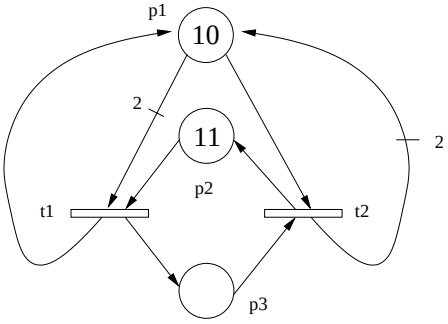

Fig. 16.6 A continuous MTS system that integrates a discrete spurious deadlock $m = [0\ 1\ 10]^T$, reachable through the firing sequence $5t_1, 2.5t_1, 1.25t_1, \ldots$.

Fortunately, some spurious deadlocks can be removed from the state equation by adding implicit places [7]. For this, it is firstly necessary to detect if a deadlock m is spurious or not. Let us define $\mathbf{Pre}_\Theta$ and $\mathbf{Post}_\Theta$ as $|P| \times |T|$ sized matrices such that:

- $\mathbf{Pre}_\Theta[p,t] = 1$ if $\mathbf{Pre}[p,t] > 0$, $\mathbf{Pre}_\Theta[p,t] = 0$ otherwise
- $\mathbf{Post}_\Theta[p,t] = |{}^\bullet t|$ if $\mathbf{Post}[p,t] > 0$, $\mathbf{Post}_\Theta[p,t] = 0$ otherwise.

Equations $\{y^T \cdot C_\Theta = 0,\ y \geq 0\}$ where $C_\Theta = \mathbf{Post}_\Theta - \mathbf{Pre}_\Theta$ define a generator of traps (Θ is a trap iff $\exists y \geq 0$ such that $\Theta = \|y\|$, $y^T \cdot C_\Theta = 0$) [13, 34]. Hence:

Proposition 10 *Given m, if the following bilinear system:*

- $m = m_0 + C \cdot \sigma,\ \ m, \sigma \geq 0,$ $\{$*state equation*$\}$
- $y^T \cdot C_\Theta = 0, y \geq 0,$ $\{$*trap generator*$\}$
- $y^T \cdot m_0 \geq 1,$ $\{$*initially marked trap*$\}$
- $y^T \cdot m = 0,$ $\{$*trap empty at m*$\}$

has solution, then m is a spurious solution, and y is the support of a trap that becomes empty.

The result of Proposition 10 follows directly from the fact that $\|y\|$ is a trap that has been emptied. Let us illustrate now how spurious solutions can be cut by adding an *implicit* place (a place is said to be *implicit* if it is never the unique place that forbids the firing of its output transitions, i.e., it does not constraint the behavior of the sequential net system).

Recalling the example of Fig. 16.6, since p_1 is an initially marked trap, its marking must satisfy $m[p_1] \geq 1$. This equation together with the conservative law $m[p_1]+m[p_3] = 10$ leads to $m[p_3] \leq 9$. This last inequality can be forced by adding a slack variable, i.e., a *cutting implicit place* q_3, such that $m[p_3]+m[q_3] = 9$. Thus q_3 is a place having t_2 as input transition, t_1 as output transition and 9 as initial marking. The addition of q_3 to the net system renders p_2 implicit (structurally identical but with a higher marking) and therefore p_2 can be removed without affecting the system behavior [7, 34]. In the resulting net system, $m = [0\ 1\ 10]^T$ is not any more a solution of the state equation, i.e., it is not lim-reachable and then the net system does not deadlock as continuous.

Since deadlock markings in continuous systems are always in the borders of the convex set of reachable markings, discrete spurious deadlocks can be cut by the described procedure. Nevertheless, such an addition creates more traps that might be treated similarly in order to improve further the quality of the continuous approximation. It is important to remark that, by eliminating spurious deadlocks, the approximation of the performance of the discrete net system, provided by the timed relaxation, is also improved even if the deadlock is not reached in the timed continuous model. In any case, removing spurious solutions always represents an improvement of the fluidization, being specially important when those are deadlocks or represent non-live steady states.

As an example, consider again the MPN given by the net of Fig. 16.6 with initial marking $M_0 = [10\,1\,1\,0]^T$ and rates $\lambda = [0.4\,1]$. As already shown, this PN has a spurious deadlock, which can be removed by eliminated the two *frozen tokens* from p_2. This is equivalent to consider $M'_0 = [10\ 9\ 0]^T$ as the initial marking. The MPN and the corresponding fluid model TCPN have been simulated for both initial markings M_0 (with spurious deadlocks) and M'_0, for different rates at t_1 ranging in $\lambda_1 \in [0.4, 4]$. The throughput at t_1, for both models, is shown if Fig. 16.7(a). It can be seen that the MPN is live for any $\lambda_1 \in [0.4, 4]$, furthermore, the throughput seems as a smooth function of λ_1. On the other hand, the continuous model with the original M_0 reaches the (spurious) deadlock for any $\lambda_1 \in (2, 4]$. Note the discontinuity at $\lambda_1 = 2$ for the TCPN model with both initial markings, i.e., the continuous model is neither monotonic nor smooth w.r.t the timing. Finally, it can be appreciated that the TCPN provides a much better approximation when the spurious deadlock is removed (TCPN$'$ with M'_0), for any $\lambda_1 > 2$ (for $\lambda_1 \leq 2$ there is no change in the TCPN).

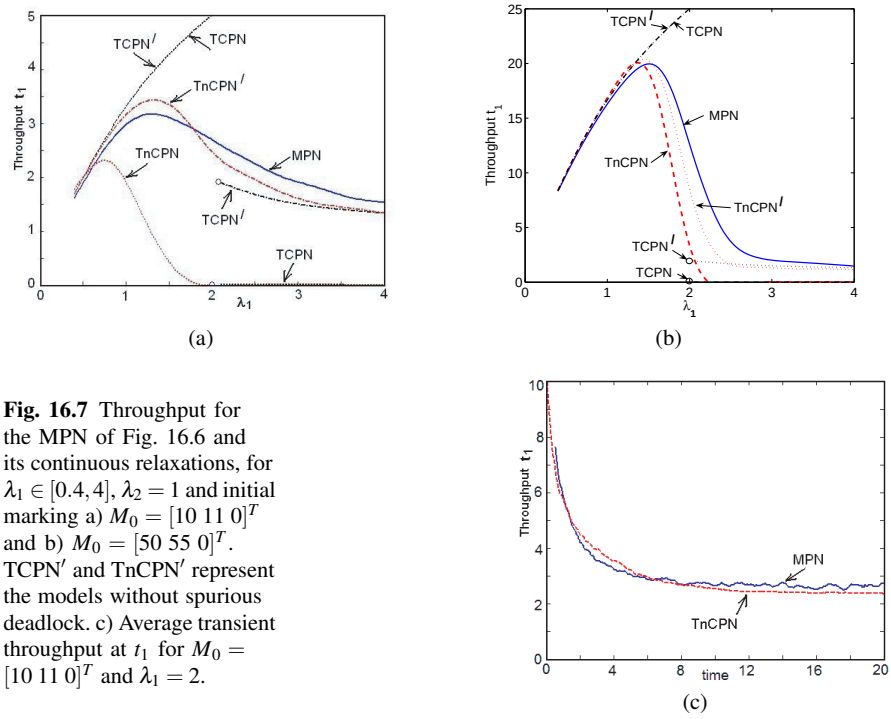

Fig. 16.7 Throughput for the MPN of Fig. 16.6 and its continuous relaxations, for $\lambda_1 \in [0.4, 4]$, $\lambda_2 = 1$ and initial marking a) $M_0 = [10\ 11\ 0]^T$ and b) $M_0 = [50\ 55\ 0]^T$. TCPN$'$ and TnCPN$'$ represent the models without spurious deadlock. c) Average transient throughput at t_1 for $M_0 = [10\ 11\ 0]^T$ and $\lambda_1 = 2$.

16.4.2 Stochastic continuous PN

The approximation of the average marking of an ergodic (ergodicity means that the steady state is independent on the initial state, providing the P-flows have the same total marking) Markovian Petri net may be improved by adding white noise to the transitions flow of the TCPN [36]. Intuitively speaking, the transition firings of a MPN are stochastic processes, then, the noise added to the flow in the TCPN may help to reproduce such stochastic behavior, which even at steady state is particularly relevant at the synchronizations. The model thus obtained (here denoted as TnCPN) is represented, in discrete time with a sampling $\Delta\tau$, as: $m_{k+1} = m_k + C(\Lambda\Pi(m_k)m_k\Delta\tau + v_k)$, with v_k being a vector of independent normally distributed random variables with zero mean and covariance matrix $\sum_{v_k} = diag(\Lambda\Pi(m_k)m_k\Delta\tau)$. This modification is particularly relevant when the system evolves near to the border between different regions, because in these cases, the continuous flow does not approximate the throughput of the discrete transitions (remember that, in a join $\{p_i^1, .., p_i^k\} = {}^\bullet t_i$, the difference between $E\{Enab(t_i)\} = E\{min(M_i^1, .., M_i^k)\}$ and its continuous approximation $enab(t_i) = min(m[p_i^1], .., m[p_i^k]) \sim min(E\{M[p_i^1]\}, .., E\{M[p_i^k]\})$ may become large). The approximation improves as the enabling degrees of the transitions (the

number of active servers) increase, as already mentioned, assuming *asymptotic stability* and *liveness* in the continuous system (thus it is important to previously remove any spurious deadlock).

An interesting issue is that the new continuous stochastic model approximates not only the average value but also the covariance of the marking of the original MPN. Moreover, since the TnCPN model is actually the TCPN one with zero-mean gaussian noise, many of the results known for the deterministic model can be used for analysis and synthesis in the stochastic continuous one. Nevertheless, the addition of noise cannot reduce the error introduced by arc weights.

For instance, consider again the MPN system of Fig. 16.5(b). The corresponding TnCPN was simulated for $\lambda_4 \in \{2, 1.5, 1.2, 1\}$. The average steady state marking is also shown in table 16.2. As it was pointed out in the previous section, the lower the probability that M_k belongs to $\mathscr{R}_{ss}$, the larger the difference (the error) between the MPN and the deterministic TCPN. On the other hand, the approximation provided by the TnCPN system is good for all of those rates.

Now, consider again the MPN of Fig. 16.6 with $M_0 = [10\,11\,0]^T$. The steady state throughput of the MPN and its different relaxations is shown if Fig. 16.7(a), for different values $\lambda_1 \in [0.4, 4]$. Note that the noise added to the TCPN makes this to reach the spurious deadlock quickly and the approximation to the MPN does not hold since the liveness precondition is not fulfilled. On the other hand, after removing the spurious deadlock with $M'_0 = [10\ 9\ 0]^T$, the TnCPN approximates better the MPN than the TCPN model (curves TnCPN$'$ and MPN). Fig. 16.7(a) shows the results of the same experiment but with a bigger population. In this case, $M_0 = 5 \cdot [10\ 11\ 0]^T = [50\ 55\ 0]^T$ and the spurious solution is removed by considering the initial marking $M'_0 = [50\ 49\ 0]^T$ (in this case, six frozen tokens are removed from p_2). Note that this marking is not equal to five times the one used in the first case, i.e., $M'_0 \neq 5 \cdot [10\ 9\ 0]^T$, then the curve TCPN$'$ in Fig. 16.7(b) is not in homothetic relation with that in Fig. 16.7(a) (but the original TCPN does it). It can be observed in Fig. 16.7(b) that now the continuous models provide a better approximation than in the case of Fig. 16.7(a), because the population is bigger. Finally, Fig. 16.7(c) shows the transient trajectory described by the average throughput of t_1, for the case $M_0 = [10\ 11\ 0]^T$ and $\lambda_1 = 2$. It can be observed, that not only the steady state of the MPN is well approximated by the TnCPN$'$ (after removing the spurious deadlock), but also the transient evolution.

16.5 Steady state: performance bounds and optimization

Product semantics may lead to continuous PN systems with steady orbits or limit cycles [30]. This semantic also allows the existence of chaotic behaviors. Analogously, when ISS are considered, a TCPN system may exhibit stationary oscillations (that can be maintained for ever). An example of an oscillatory behavior, can be found in [19] (Fig. 2 and 3). Usually, a TCPN evolves toward a steady state marking, like in the examples of Section 16.3. The knowledge of this final marking is interesting

for performance evaluation purposes, since this represents the number of useful resources and active clients in the long term behavior of the modeled system. This is explored through this section.

A performance measure that is often used in discrete PN systems is the throughput of a transition in the steady state (assuming it exists), i.e., the number of firings per time unit. In the continuous approximation, this corresponds to the firing flow.

In order to study the throughput in discrete systems, the classical concept of "visit ratio" (from the queueing network theory) is frequently used. In Petri net terms, the visit ratio of a transition t_j with respect to t_i, $v^{(i)}[t_j]$, is the average number of times t_j is visited (fired), for each visit to (firing of) the reference transition t_i.

Let us consider consistent nets without empty siphons at m_0 (Prop. 5). In order to simplify the presentation, let us assume that the net is MTS. Therefore, for any t_i, $f_{ss} = \chi_i \cdot v^{(i)}$, with χ_i the throughput of t_i. The vector of visit ratios is a right annuler of the incidence matrix C, and therefore, proportional to the unique T-semiflow in MTS systems. For this class of systems, a throughput bound can be computed using the following non-linear programming problem that maximize the flow of a transition (any of them, since all are related by the T-semiflow)

$$\begin{array}{ll} \max & f_{ss}[t_1] \\ \text{s.t.} & m_{ss} = m_0 + C \cdot \sigma \\ & f_{ss}[t_j] = \lambda_j \cdot \min\limits_{p_i \in {}^\bullet t_j} \left\{ \frac{m_{ss}[p_i]}{Pre[p_i,t_j]} \right\}, \forall t_i \in T \\ & C \cdot f_{ss} = 0 \\ & m_{ss}, \sigma \geq 0 \end{array} \tag{16.7}$$

where m_{ss} is the steady-state marking. A way to solve (16.7), that due to the minimum operator is non linear, consists in using a branch & bound algorithm [17]. Relaxing the problem to a LPP, an upper bound solution can be obtained in polynomial time, although this may lead to a non-tight bound, i.e., the solution may be not reachable if there exists a transition for which the flow equation is not satisfied. If the net is not MTS, similar developments can be done adapting the equations in [6].

In the case of controlled systems, a LPP transformation of (16.7) can be used to compute an optimal steady-state assuming only flow reduction (the speed of the machines can only be reduced), $f \geq 0$ and the steady-state flow should be repetitive $C \cdot f = 0$. If all transitions are controllable, it can be solved introducing some *slack* variables in order to transform the inequalities derived from the minimum operator in some equality constraints. These slack variables are used after to compute the optimal steady-state control [19]. For example, let us consider the following LPP:

$$\begin{array}{ll} \max & k_1 \cdot f - k_2 \cdot m - k_3 \cdot m_0 \\ \text{s.t.} & C \cdot f = 0, f \geq 0 \\ & m = m_0 + C \cdot \sigma, m, \sigma \geq 0 \\ & f_i = \lambda_i \cdot \left(\frac{m[p_j]}{Pre[p_j,t_i]} \right) - v[p_j,t_i], \forall p_j \in {}^\bullet t_i, v[p_j,t_i] \geq 0 \end{array} \tag{16.8}$$

where $v[p_j,t_i]$ are *slack* variables. The objective function represents the profit that has to be maximized where k_1 is a price vector w.r.t. steady-state flow f, k_2 is the

work in process (WIP) cost vector w.r.t. the average marking m and k_3 represents depreciations or amortization of the initial investments over m_0. Using the slack variables v the optimal control in steady-state for a transition t_i if it is controllable, i.e., permits a control $u_i > 0$, is just $u_i = \min_{p_j \in {}^\bullet t_i} v[p_j, t_i]$. Therefore, this control problem (a synthesis problem) seems easier than the computations of performance (an analysis problem) even if, in general, is the opposite. Controllability issues will be considered from dynamic perspective in chapter 19.

References

1. E. Altman, T. Jimnez, and G. Koole. On the comparison of Queueing Systems with their Fluid limits. *Probability in the Engineering and Informational Sciences*, 15:165–178, 2001.
2. F. Balduzzi, A. Giua, and G. Menga. First-Order Hybrid Petri Nets: a Model for Optimization and Control. *IEEE Trans. on Robotics and Automation*, 16(4):382–399, 2000.
3. G. W. Brams. *Réseaux de Petri: Théorie et Pratique*. Masson, 1983.
4. R. Champagnat, P. Esteban, H. Pingaud, and R. Valette. Modeling and simulation of a hybrid system through pr/tr pn-dae model. In *Proceedings of ADPM98*, pages 131–137, 1998.
5. H. Chen and A. Mandelbaum. Discrete flow networks: Bottleneck analysis and fluid approximations. *Mathematical Operations Research*, 16:408–446, 1991.
6. G. Chiola, C. Anglano, J. Campos, J. M. Colom, and M. Silva. Operational Analysis of Timed Petri Nets and Application to the Computation of Performance Bounds. In F. Baccelli, A. Jean-Marie, and I. Mitrani, editors, *Quantitative Methods in Parallel Systems*, pages 161–174. Springer, 1995.
7. J. M. Colom and M. Silva. Improving the Linearly Based Characterization of P/T Nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 113–145. Springer, 1991.
8. J. G. Dai. On positive Harris recurrence of multiclass queuing networks: A unified approach via fluid limit models. *The Annals of Applied Probability*, 5(1):49–77, 1995.
9. R. David and H. Alla. Continuous Petri Nets. In *Proc. of the 8th European Workshop on Application and Theory of Petri Nets*, pages 275–294, Zaragoza, Spain, 1987.
10. R. David and H. Alla. Autonomous and timed Continuous Petri Nets. In *Proc. 11th Int. Conf. on Application and Theory of Petri Nets*, pages 367–386, Paris, France, 1990.
11. R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2004. (Revised 2nd edition, 2010).
12. I. Demongodin, N. Audry, and F. Prunet. Batches Petri nets. In *IEEE conference on Systems, Man and Cybernetics, 'Systems engineering in the service of humans'*, pages 607–617, 1993.
13. J. Ezpeleta, J. M. Couvreur, and M. Silva. A New Technique for Finding a Generating Family of Siphons, Traps and ST-Components. Application to Coloured Petri Nets. In G. Rozenberg, editor, *Advances in Petri Nets 1993*, volume 674 of *Lecture Notes in Computer Science*, pages 126–147. Springer, 1993.
14. M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. *Formal Methods for Computational Systems Biology*, pages 215–264, 2008.
15. J. Hillston. Fluid Flow Approximation of PEPA models. In *Procs. of the Second Int. Conf. on the Quantitative Evaluation of Systems (QEST). IEEE Computer Society.*, pages 33–43, 2005.
16. J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In W. van der Aalst and E. Best, editors, 24th *Int. Conf. on Application and Theory of Petri Nets (ICATPN 2003)*, volume 2679 of *Lecture Notes in Computer Science*, pages 221–240. Springer, The Netherlands, June 2003.
17. J. Júlvez, L. Recalde, and M. Silva. Steady state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605–616, May 2005.

18. J. Júlvez, L. Recalde, and M. Silva. Deadlock-freeness analysis of continuous mono-t-semiflow petri nets. *Automatic Control, IEEE Trans. on*, 51(9):1472–1481, Sept. 2006.
19. C. Mahulea, A. Ramírez, L. Recalde, and M. Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Transactions on Automation Science and Engineering*, 5(2):307–320, 2008.
20. C. Mahulea, L. Recalde, and M. Silva. Basic server semantics and performance monotonicity of continuous Petri nets. *Discrete Event Dynamic Systems*, 19(2):189–212, 2009.
21. C. Mahulea, L. Recalde, and M. Silva. Observability of continuous Petri nets with infinite server semantics. *Nonlinear Analysis: Hybrid Systems*, 4(2):219–232, 2010.
22. M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with generalized stochastic Petri nets*. John Wiley and Sons, 1995.
23. L. Recalde, S. Haddad, and M. Silva. Continuous Petri Nets: Expressive Power and Decidability Issues. *Int. Journal of Foundations of Computer Science*, 21(2):235–256, 2010.
24. L. Recalde and M. Silva. Petri nets fluidification revisited: Semantics and steady state. *European Journal of Automation APII-JESA*, 35(4):435–449, 2001.
25. L. Recalde and M. Silva. Petri Nets Fluidification revisited: Semantics and Steady state. *APII-JESA*, 35(4):435–449, 2001.
26. L. Recalde, E. Teruel, and M. Silva. On Linear Algebraic Techniques for Liveness Analysis of P/T Systems. *Journal of Circuits, Systems, and Computers*, 8(1):223–265, 1998.
27. L. Recalde, E. Teruel, and M. Silva. Autonomous Continuous P/T systems. In J. Kleijn S. Donatelli, editor, *Application and Theory of Petri Nets 1999*, volume 1639 of *Lecture Notes in Computer Science*, pages 107–126. Springer, 1999.
28. M. Silva. Towards a Synchrony Theory for P/T Nets. In K. Voss et al., editors, *Concurrency and Nets*, pages 435–460. Springer, 1987.
29. M. Silva and J.M. Colom. On the structural computation of synchronic invariants in P/T nets. In *Proc. of the 8th European Workshop on Application and Theory of Petri Nets*, pages 237–258, Zaragoza, Spain, 1987.
30. M. Silva and L. Recalde. Petri nets and integrality relaxations: A view of continuous Petri nets. *IEEE Trans. on Systems, Man, and Cybernetics*, 32(4):314–327, 2002.
31. M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28:253–266, 2004.
32. M. Silva and E. Teruel. A Systems Theory Perspective of Discrete Event Dynamic Systems: The Petri Net Paradigm. In P. Borne, J. C. Gentina, E. Craye, and S. El Khattabi, editors, *Symp. on Discrete Events and Manufacturing Systems. CESA '96 IMACS Multiconference*, pages 1–12, Lille, France, July 1996.
33. M. Silva and E. Teruel. DEDS Along Their Life Cycle. Interpreted Extensions of Petri Nets. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, La Jolla, San Diego, CA, USA, October 1998.
34. M. Silva, E. Teruel, and J. M. Colom. Linear Algebraic and Linear Programming Techniques for the Analysis of Net Systems. In G. Rozenberg and W. Reisig, editors, *Lectures in Petri Nets. I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 309–373. Springer, 1998.
35. K. Trivedi and V. G. Kulkarni. Fspns: Fluid Stochastic Petri nets. *Application and Theory of Petri Nets*, *691 of Lecture Notes in Computer Science*:24–31, 1993.
36. C. R. Vázquez, L. Recalde, and M. Silva. Stochastic–continuous state approximation of Markovian Petri net systems. In *47th IEEE Conference on Decision and Control*, 2008.
37. C. R. Vázquez and M. Silva. Timing and liveness in continuous Petri nets. *Automatica*, 47:283–290, 2011.
38. C.R. Vázquez and M. Silva. Timing-dependent boundedness and liveness in continuous Petri nets. In *10th Int. Workshop on Discrete Event Systems (WODES)*, Berlin, Germany, 2010.