On Fluidization of Discrete Event Models: Observation and Control of Continuous Petri nets
-draft-

Manuel Silva, Jorge Júlvez, Cristian Mahulea and Renato Vázquez*

December 23, 2014

Abstract

As a preliminary overview, this work provides first a broad tutorial on the fluidization of discrete event dynamic models, an efficient technique for dealing with the classical *state explosion* problem. Even if named as continuous or fluid, the relaxed models obtained are frequently hybrid in a technical sense. Thus, there is plenty of room for using discrete, hybrid and continuous model techniques for logical verification, performance evaluation and control studies. Moreover, the possibilities for transferring concepts and techniques from one modeling paradigm to others are very significant, so there is much space for synergy. As a central modeling paradigm for parallel and synchronized discrete event systems, Petri nets (PNs) are then considered in much more detail. In this sense, this paper is somewhat complementary to [49]. Our presentation of fluid views or approximations of PNs has sometimes a flavor of a survey, but also introduces some new ideas or techniques. Among the aspects that distinguish the adopted approach are: the focus on the relationships between *discrete* and *continuous* PN models, both for *untimed*, i.e., fully non-deterministic abstractions, and *timed* versions; the use of *structure theory* of (discrete) PNs, algebraic and graph based concepts and results; and the bridge to Automatic Control Theory. After discussing observability and controllability issues, the most technical part in this work, the paper concludes with some remarks and possible directions for future research.

Published as:
M. Silva, J. Julvez, C. Mahulea, C.R. Vazquez, "On fluidization of discrete event models: observation and control of continuous Petri nets," Discrete Event Dynamic Systems: Theory and Applications, vol. 21, no. 4, December 2011.. DOI: `http://doi.org/10.1007/s10626-011-0116-9`

*The authors are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain, {*silva,julvez,cmahulea,cvazquez*}@*unizar.es*.

1 Introduction

Real systems are not continuous, discrete or hybrid. Continuous, discrete or hybrid are the *models* that we construct in order to represent them. According to the given purposes, a system can be "viewed" at different moments through different models, particularly during its *life-cycle*. In fact, models are always abstractions! For example, contrary to what is usually claimed, a tank filled with pure water *is not* a continuous system, because we know that the liquid is formed by molecules, so there is always a discrete number (remember, nanotechnologies deal with matter on an atomic and molecular scale). Moreover, molecules are formed by atoms. But XX century physics tells us that atoms "can be divided" and it becomes difficult to classify the models at the atomic level as discrete, continuous or hybrid in the classical sense[1]. In the same line of thinking, classical predator-prey models (such as the basic model of Volterra-Lotka) are based on a fluid view of systems in which the number of individuals is discrete (even if many additional but abstracted features may distinguish them). As a last example, among many, when visiting a large manufacturing plant, it is common to hear the engineers using a kind of hydraulic terminology: *levels, flows*, etc. Having said that, it should be noted that, by abuse of language, expressions such as continuous, discrete or hybrid "systems" are lexicalized, but refer to models, i.e., they are "views". According to this accepted practice, models and systems are substantives frequently used interchangeably, even if sometimes a precision is truly needed. Engineers are more interested in pragmatism than ontology (essences), and the manipulation of "fluid (or continuous) views" of systems is a useful and classical approach.

The mathematics for continuous dynamic systems, particularly for control, goes back more than three centuries [160]. At the intersection of Automatic Control, Operations Research and Computer Science, the formalization of discrete event dynamic "views" is more recent. Very roughly speaking it can be said that such views were really developed during the second half of the past century[2].

In many human made systems, for example in telecommunications, manufacturing, logistics, transportation, work-flow management or distributed computation, the conceptually "more appropriate" kind of formal representation belongs to the class of *Discrete Event Dynamic Systems* (DEDS). But this "natural" formalization may suffer from the well known *state explosion problem*. Then transformation and structural techniques (where the initial conditions play the role of a parameter) may be of interest, but they do not offer complete solutions for all imaginable cases. One of the more simple and classical "fluid" relaxation is that of transforming Linear Integer Programming Problems (IPP, computationally NP-hard) into Linear Programming Problems (LPP, of polynomial time complexity), a proof that *to fluidize* has been in general an invaluable intellectual resource to construct more abstract or coarse models. The main goal is to make computational problems for large-scale systems decidable or much more tractable.

Fluidization techniques try to obtain semi-decision or approximate solutions for

[1]Think, for example, in models of atoms based on wave-particle duality or the uncertainty principle (Bernard Pullman, *The Atom in the History of Human Thought*, Oxford Univ. Press, 1998).

[2]Before electronic computers became a reality, it is clear that the foundations of Theoretical Computer Science go back to the 1930s with the works of Alan Turing. Moreover, in 1928 Claude Shannon applied Boolean Algebra while developing Switching Theory. Even some years before, in the electromechanical domain, Leonardo Torres Quevedo's chess player was driven by an automata. Furthermore, it is clear that, for example, the pioneering works of Markov and Erlang belong to the first decades of the XX century. Nevertheless, Automata Theory or Queueing Theory (Operations Research in a broader sense), are identifiable bodies of literature defined by the foundational research of the 1950's and 1960's.

qualitative and quantitative analysis of complex models, instead of exact solutions of (over)simplified "views" of the original system. Fortunately, the quality of the approximation usually improves with the size of the population in the system being considered, while at the same time the computational savings with respect to the discrete underlying model become much more significant.

Frequently the fluid approximation provides some useful insights into the potential behavior of the real system under consideration, even if the size of the populations is not "too big". In other words, fluidization may allow to get tractable approximate solutions to be achieved by providing some "educated guess" about behaviors (instead of a very precise evaluation of an artificially over simplified model). Anyhow it is particularly relevant to remember once again that, in essence, any model is just an approximation of a certain reality. Obviously, part of the price paid for fluidization is that, in general, certain properties cannot be considered in a fluid framework, for example, mutual exclusion.

Coarsening by fluidization is a frequently practised relaxation technique, but there are several others. Among these, *decomposition techniques* (the idea of "divide and conquer") or *Lagrangian relaxations*, which employ duality properties of dynamic systems, especially useful in optimization problems. Rather than alternative, these latter relaxation techniques may be seen as complementary approaches in the struggle against computational complexity.

A *formalism* is a conceptual framework that enables a kind of formal model of systems to be obtained, allowing some mathematically-based techniques for the specification, development and verification. Its use in engineering is grounded on the expectation of contributing to the quality and robustness of a design by performing appropriate mathematical analysis. For instance, ordinary differential equations constitute a formalism for the modeling of the dynamic behavior of continuous models with lumped parameters. In view of the long life cycle of a given system (during which it is conceived, analyzed from different perspectives, implemented and operated), and the diversity of application domains, it seems desirable to have a family of formalisms rather than a collection of unrelated or weakly related formalisms. Following Thomas Kuhn's ideas (*The Structure of Scientific Revolutions*, 1962), a *paradigm* is "the total pattern of perceiving, conceptualizing, acting, validating, and valuing associated with a particular image of reality that prevails in a science or a branch of science". For us, a *modeling paradigm* is a conceptual framework that allows formalisms to be obtained from some common concepts and principles, with the consequent economy, coherence and synergy in development, among other benefits.

In this paper, fluid or continuous Petri nets are not seen as isolated formalisms, but as part of a broad modeling paradigm for DEDS, the *Petri nets paradigm* [156, 157]. Based on the expression of *concurrency* and *synchronization*, locality of states (places) and actions (transitions) is a basic issue for Petri nets. One of the main consequences is the possibility of approaching the design, analysis and implementation of parallel and synchronized, eventually distributed, systems using *bottom-up* (by composition of lower level modules) and *top-down* (by refinement of very abstract or upper level descriptions) approaches, in an arbitrary interleaved manners. Petri Net (PN) models can then be presented as *flat* or *structured* descriptions, in the latter case just by keeping track of the way they are constructed, a degree of freedom of the modelers and users. Structuring is in any case an essential issue when dealing with the modeling and analysis of complex systems.

Among the most cited advantages of PN models is their representability in graphical terms, but it is improper to limit them to a graphical formalism because they can be also described straightforwardly in a textual way, which may be convenient for very large models. Useful for the modeling, analysis and synthesis of concurrent and distributed systems formalized as discrete [22, 49, 89, 136, 147, 149, 163], the conceptual centrality of PNs in the framework of DEDS is confirmed when considering that they have been defined from quite different perspectives [156]: axiomatically (by C. A. Petri himself), through the Vector Addition System, through the Theory of Regions of a labeled graph (encoding the set of nodes-global states), or from Linear Logic (non-monotonic logic of Girard), to give some examples.

The introduction of *continuous Petri nets* (CPNs) goes back to 1987 [47]. R. David explicitly states (see [49], p. IX) that the source of inspiration was the fluidization of models for the performance evaluation of production lines. It is simply coincidence that, at the same meeting in Zaragoza a systematic use of Linear Programming techniques for the structural analysis of Petri nets was proposed, working with the *fundamental* or *state equation* of the net system [151] (a revised version in [26]). In fact this second proposal can be rephrased as just relaxing Integer Programming Problems into Linear Programming Problems in order to obtain necessary or sufficient conditions for qualitative properties, or bounds for quantitative ones. The main difference between both approaches is that, being more conceptual, R. David and H. Alla fluidize at the net level, while in the second case, being more technical, fluidization is applied at the level of equations. In perspective, the advantage of the approach of our colleagues from Grenoble was the possibility of describing the transient behavior of timed models, a topic in which we did not take an interest until around a decade later. On the other hand, the advantage of our approach is that the possible consideration of the fundamental equation in the integer domain may be very important in order to improve the analysis of the discrete event models.

Fluidization of PN models is mainly considered here at the level of transitions, leading to the fluidization of places in their pre- and post-conditions. When only some transitions are fluidized, the PN model is said to be *hybrid* [49, 53]. Close to this idea, a different class of hybrid PNs has been called *Fluid PNs* [86, 162], a formalism in which the marking of some places is relaxed into the non-negative real numbers in the framework of a stochastic model. In general, partial relaxations are used in very diverse conceptual frameworks and application domains, for example, in data communication networks. Their packet-level granularity is sometime abstracted into a packet-train granularity, i.e., clusters of closely-spaced packets are replaced by "packets trains" (see, for example, [116]). In the cited framework of PNs, hybrid abstractions may be used, for example, to represent "platoons" of vehicles in road traffic problems, these being formed due to the synchronization imposed by traffic lights [169]. In order to deal with similar problems, *Batches PNs* were defined for the modeling and analysis of bottling lines [50, 51]. In this case, the previous kind of formalism is enriched with additional primitives. *First-Order Hybrid Petri Nets* (FOHPNs) represent an alternative definition of a timed PN based hybrid formalism. Using LPP techniques, in [11] some on-line control and structural optimization problems are considered; even a multi-class production network described with a queueing network is considered in the FOHPNs framework. Another hybrid PN extension is *Differential Petri Nets*, which admit negative-real markings [52].

If all transitions of a discrete PN are fluidized, the net model is said to be *fluid* or *continuous*, but even in that extreme case the formalism is most frequently technically hybrid. In this work we concentrate on *fluid* or *continuous* Petri Nets, formalisms that are particularly appropriate for modeling many Large Scale Networks. Moreover, a good understanding of continuous PNs is a fundamental issue for improving the understanding of most hybrid PN formalisms. Nevertheless, certain PN based hybrid formalisms use a different approach to incorporate some continuous part, particularly *Differential Predicate Transitions PNs* [33, 34]. Using the same kind of approach employed in *hybrid automata* (see, for example, [4]), the essential difference is that the discrete dynamics are described with Petri nets.

The present work is structured as follows: In section 2 a broad panorama is traced, presenting a few important paradigms used for formal modeling and analysis of large and distributed DEDS. One of the goals is to highlight the fact that despite the apparent diversity at first sight, there exist many common features, and there are many possibilities of enriching the different perspectives through the incorporation or adaptation of concepts and techniques initially developed in other paradigms. Fortunately, this transfer of concepts and techniques is particularly interesting in the case of derived fluid models. Sections 3 and 4 introduce *continuous Petri Nets* (CPNs) as untimed, i.e., fully non-deterministic, and timed formalisms, respectively. The relationship between the properties of (discrete) PNs and the corresponding properties of their continuous approximation is considered at several points. Even if fluidization leads to more efficient techniques for analysis, it should be emphasized that the expressive power of timed CPNs (TCPN) (under infinite server semantics) is surprisingly high, because they can simulate Turing Machines [138]. This means that certain important properties such as marking coverability or the existence of a steady-state are undecidable.

Among the main technical problems that we review are the observation (section 5) and control (section 6) of TCPNs. In particular, a blend of techniques belonging to PNs and (continuous and hybrid) Automatic Control theories is used, emphasizing some structural (graph and algebraic) concepts and results. The control problems considered are mainly of the *set-point* or *state-targeting* type (where the distributed state is the marking), or *state-tracking* control. For example, if the time duration is minimized in the transfer from an initial to a given state, the problem is analogous to the *scheduling* problem in which the goal is to minimize the *make span* in the corresponding discrete model. Enforcement of some safety constraints, e.g., deadlock-freeness or generalized mutual exclusion constraints, may be previously considered using PN based techniques. This overview ends with some concluding remarks (section 7).

2 Fluidization: a common coarsening approach for different DEDS modeling paradigms

The purpose of this section is to provide a quick, probably over-ambitious, overview of the field. More than two decades ago, in the Fleming report about *Future directions in control theory: a mathematical perspective*, it was stated that [63]:

> there exist no formalisms for DEDS mathematically as compact or computationally as tractable as the differential equations are for continuous systems, particularly with the goal of control.

Certainly the field is much more mature today, as it can be readily verified by looking at the thousands of published works and their applications to real problems. Nevertheless, it can be said that the same basic operational formalisms remain today, and considerable diversity still prevails in the DEDS arena. Therefore, the idea in this section is to "open the window" in order not to limit the perspective to the Petri nets paradigm, but to show fluidization as a broad tendency. The goal is to highlight the existence of similarities and potential synergies among different modeling paradigms, and to suggest that many concepts and techniques can be borrowed from one modeling paradigm to improve or approach others.

Fluid approximations of discrete event dynamic models may be obtained as a limit case through a continuous state relaxation of the discrete model. Roughly speaking, the coarse representation given by fluid models is obtained by abstracting the movement of discrete entities: the new focus deals with the change of the aggregated flows, the use of hydrodynamic metaphors being frequent (stock-levels and flows). This leads to reasonable results when the loads are large enough and the stochastic fluctuations may be neglected in relative terms, as it may happen, for example, under heavy traffic conditions. Therefore, the relation between the discrete model and its relaxed approximation is an important topic. Alternatively, fluid models may also be directly introduced without paying much attention to the possible relation with an underlying discrete model, i.e., assuming from the very beginning, at the modeling stage, that for the problem being considered a fluid model provides a "good enough abstract view" of the expected behaviors. The first set includes fluid formalisms derived from *Queueing Networks* (QNs, sec. 2.1), *Stochastic Petri Nets* (SPNs) or *Markovian Process Algebra* (MPAs, sec. 2.3). The second group comprises other well-known formalisms such as *Stochastic Flow Models* (SFMs) or *Forrester Diagrams* (FDs, also expressively called *Stock and Flow Diagrams*) (sec. 2.2). In all cases, considering very simplified historical traces, PNs will be considered as a co-existing paradigm. Thus, no special subsection is explicitly devoted to them here. Nevertheless, they are usually taken as a reference, while we emphasize the convenience of dealing with multi-paradigm views.

2.1 Queueing Networks and fluid views

The history of Queueing Theory goes back to the beginning of the XX century with the pioneering works of A. K. Erlang for telecommunication networks. Queues were defined originally in order to deal with *resource contention* among independent jobs, e.g., problems of congestion in traffic engineering. Pioneering works on Queueing Networks go back to the late 1950s (J. Jackson, 1957) and the beginning of the sixties.

In parallel, Petri Nets were introduced by C. A. Petri at the beginning of the 1960s [137], as a fully *non-deterministic* (untimed) conceptual framework to logically model and analyze *concurrency* and *synchronization* in DEDS. Perceived as a System Theory by Petri (initially axiomatic), this field was called *Systemics* by A. Holt in the historic MAC Project of MIT [60], and the marked bipartite diagrams representing the systems were baptized *Petri Nets*. Until the mid 1970s, this was mainly related to the framework of parallel programming. Notions of time in order to compute performance or dependability were added to PNs around two decades later, at the beginning of the 1980s. In the PN paradigm, time has been introduced for different purposes (for example, *time intervals* in order to deal with some qualitative or quantitative real-time properties [126]; or in a possibilistic fashion to handle uncertainty, or preferences, using *fuzzy sets* [29]). For performance evaluation and optimization, different semantics have also been defined, providing the probability density functions (pdfs) for service times and defining some probabilistic routing (even under some fairness constraints). The most current practice is to assume random policies for queues (places), services (usually associated to transitions that become stations, were servers operate) and routing (at conflicts) (for more elaborate disciplines, see [1]).

Since the eighties, the evolutions of QN and PN theories and applications have in part been such that they simultaneously address an increasingly overlapping class of problems. Nevertheless, it is very important to state that from a historical perspective the conceptual driving forces have been rather different. It can be said that from the beginning QNs focused on providing high level primitives that simultaneously supply the user with simple yet expressive building blocks and restrict the models that can be specified to those that can be analyzed in a relatively efficient way [171]. Very roughly speaking, in a sense QNs develop in a *bottom-up* approach. One important limitation of the initial proposals of QNs was the absence of a general construct to deal with synchronizations. However at the end of the 1970s, with the emergence of parallel and distributed systems, the need for synchronization became evident. For example, to describe *cooperation* relationships, such as the assembly of parts A and B into a new reality or par-begin/par-end constructs, and *competition* relationships, such as the sharing of servers among two different production lines, or the need for several resources in order to advance production. The reaction was to provide a bunch of specific, *ad hoc* extensions (for example, including diverse forms of synchronization in *Extended Queueing Networks*, EQNs [20, 70, 143]). Proceeding in this way, EQNs began to use a variety of specific primitives to handle synchronizations and resource constraints, sometimes with a clear redundancy in basic objects. For example, *passive resources* and *customers* reside in different kinds of nodes, or different nodes are used to model a *join* and a *resource acquisition* (see fig. 1). Finally, let us remark that stochastic Petri nets with weighted arcs, i.e., non-ordinary nets, can be used for the modeling of *bulk arrivals* and *bulk services* [105] with deterministic size of batches (given by the weights of the arcs).

In this respect, it can be pointed out that a great diversity and specificity of primitives may be convenient in order to develop concise and possibly elegant models, but this abundance tends to make formal reasoning and theory construction difficult. An ideal solution to conciliate reasoning capabilities and practical expressivity consists in having a minimal number of basic primitives in terms of which richer derived ones can be constructed. In contrast to QNs, the basic PN formalism is quite austere: only two simple and somewhat orthogonal primitives are identified (transitions and places). In a sense, it can be said that PN theory develops in a *top-down* way, giving primarily attention to the logical properties of fully non-deterministic models. In other words, PN formalisms are based on a few basic principles leading to great descriptive power, while QNs are based on several expressive blocks (with rich semantics, dealing with relatively sophisticated queues and server disciplines and also routing policies), a catalogue that increases under a more problem oriented perspective, resulting from practical needs. An initial comparison of QNs and timed PNs models for performance evaluation is provided in [171], were emphasis is put on *notation* (what models can be expressed and suitability for representing a class of models) and *evaluation efficiency* (what can be computed and computational effort required).

In historical terms, the bridges between these two overlapping families of models for performance evaluation have been fruitful. For example, synchronization has been introduced in QNs in a more restricted but systematic way when dealing with *Fork Join QN with Blocking* (FJQN/B, a class of models with the structure of strongly connected *Marked Graphs*, a well-understood subclass of PNs) [46], or to derive performance solution techniques on the PN side by adapting techniques from the QN field. The latter include performance bounds, mean value analysis, response time approximations or tensor based computations [10]. The same efficiency can be expected for their fluid approximations: timed fluid PNs may benefit from borrowing concepts and techniques from fluid QNs, while the problems in fluid PNs may also influence developments in fluid QNs.

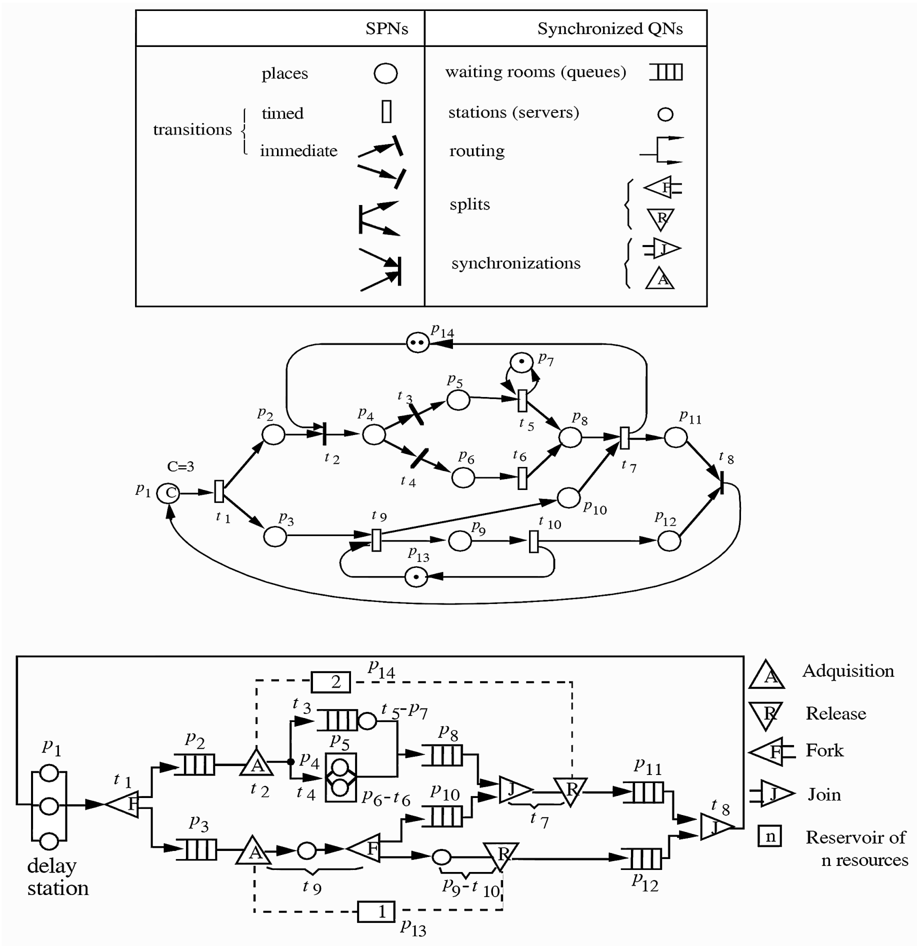


Figure 1: A stochastic Petri net and the corresponding extended queueing network

The first book dealing with fluid approximations of QNs was by G.F. Newell [133]. Recognizing that queueing theory was originated to deal with practical problems and that the literature was already very large, the preface of the book stated that:

> . . . as a tool for analysis of practical problems, it remains in a primitive state; perhaps mostly because the theory has been motivated only superficially by its potential applications... Queueing theory became very popular, particularly in the late 1950s, but its popularity did not center so much around its applications as around its mathematical aspects... The literature grew from “solutions looking for a problem” rather than from “problems looking for a solution”.
>
> Mathematicians working for their mutual entertainment will discard a problem either if they cannot solve it, or being soluble it is yet trivial. An engineer concerned with the design of a facility cannot discard the problem... he must do the best he can. The practical world of queues abounds with problems that cannot be solved elegantly but which must be analysed nev-

ertheless. The literature on queues abounds with "exact solutions", "exact bounds", simulations models, etc.; with almost everything except common sense methods of "engineering judgment".

These strong statements written in 1971 can be understood to have the intention of promoting (forty years ago!) an interest in fluid approximations for QN systems! The general acceptance of fluidization by the QN community was a slow process, being delayed some two decades following Newell's basic proposition.

Fluid limit equations can be derived from discrete QN equations by using natural extensions of the well-known law of large numbers and the central limit theorem from probability theory. More precisely its analogues in stochastic processes theory are the *Functional Strong Laws of Large Numbers* and, dealing with distributions, the *Functional Central Limit Theorem* (also known as *Donsker Theorem*). In both cases, the idea is to study the convergence of a sequence of stochastic processes to another stochastic process, generating simple approximations (see, for example, [174]). These approximations have been used in order, for example, to compute performance, analyze stability, or optimize the behavior of QNs models.

Fluid approximations may be perceived as being, in a general sense, (continuous or hybrid) dynamical systems associated with QNs. The fluid relaxed model may be *deterministic* only if the functional strong law of large numbers or similar results are used [36]. Roughly speaking, in this case arrival and service processes are replaced by their intensities, i.e., expectations or average values. Nevertheless, even if providing a simplified view, the deterministic approximation may exhibit very complex behaviors. For example, as pointed out in [23], deterministic fluid QN models need not have unique solutions, since they might bifurcate. Moreover, as will be mentioned later, when interpreting some continuous PNs as fluid EQNs, *undecidability* issues can even appear!

Among the topics considered in the QN literature, questions related to the quality of the approximation are important. When fluctuations cannot be neglected with respect to average values (for example, because the population is not truly that "big"), the fluid model should be described in terms of *stochastic differential equations*. In this latter case the noise in the differential equations partially reflects the stochastic variability in the behavior of the original discrete QN. These stochastic differential equations may be obtained by means of the functional central limit theorem or similar results.

The literature on fluid QNs has been very extensive since the 1970's. While a complete overview is outside the scope of this work, we would refer as examples to various books [23, 37, 104, 106] or articles [3, 45, 128, 129]. The parametric optimization and dynamic control (sometimes referred to as *scheduling* for the underlying discrete model) of the fluid approximate model are important problems (see for example [44, 109, 117, 131]). Among the many potential interests of fluid models is the analysis of the *stability* of discrete QNs (in PN terms, the idea of boundedness). This is a topic that in the mid 1990s was already stated to have (in certain cases) "achieved a striking success by providing a complete answer to the question of stability of stochastic networks" (page 4 in [9]), frequently irrespective of the particular discipline being applied to the queues. There is a significant amount of works dealing with fluid approaches and *stability* (among those non previously cited, for example [19, 58, 66]).

2.2 Direct fluid models of systems that can "naturally" be seen as DEDS

In many natural and technological systems the consideration of discrete event models may be conceptually the more faithful view. Nevertheless, models being abstractions, fluid views may be better suited for stability analysis, performance evaluation, sensitivity analysis or optimization, to give some examples. As already said, fluid models may be deterministic (first-moment approximation) or stochastic. Nevertheless, the basic pattern of the systems of equations is organized around the simple "mass balance or accounting principle": *the rate of accumulation, e.g., of customers in queues, is equal to the difference of incoming and outgoing flows*.

Even if conceptually thought of as continuous, those models are usually technically hybrid (among other reasons, because of upper bounds on capacities or non-negativity at the level of "reservoirs"). From a System Theory point of view, the peculiarities of models are in the definition of the input and output flows. Among an infinity of possibilities, they can be linear, piece-wise constant (as it occurs with CPN under *finite* server semantics), piece-wise linear (as occurs with CPNs under *infinite* server semantics), or bilinear (as frequently occurs in population dynamic problems, where products of predators and preys appear, a server semantics that appears in CPN by the discoloring of colored PNs [153]).

It is easy to see that in many cases, "similar" kinds of approaches are taken as the basis for defining classes of successful models of dynamical systems. For example, *Compartmental Systems* (CS) are composed of a finite number of subsystems (compartments), interacting by exchanging nonnegative quantities of material and energy among the compartments and with the environment [17, 172]. Similar to that of QNs but "from the beginning" a continuous perspective, compartmental "views" of systems are used in biology, medicine and ecology, among many other application fields. The system is governed by laws of transfer and conservation, while the state variables are constrained to remain nonnegative over the system trajectories. A compartmental system can be represented as a graph. A *Compartmental Network* (CN) has compartments as nodes, and has a peculiar interpretation associated to it. The *level* (or amount of material) of each compartment, x_i, changes according to the input and output flows through the arcs, i.e., $\dot{x}_i = \sum_k f_{ki} - \sum_j f_{ij}$.

In compartmental systems, generation of matter is forbidden. Hence, in the case of *linear* systems ($\dot{\boldsymbol{x}}(t) = \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{B} \cdot \boldsymbol{u}(t)$), all the eigenvalues of $\boldsymbol{A}$ have a non-positive real part and so the systems are either asymptotically or marginally stable. If it is a closed system, then $\mathbf{1} \cdot \boldsymbol{A} = \mathbf{0}$ (thus, $\boldsymbol{A}$ is singular), and then $\mathbf{1} \cdot \boldsymbol{x}(t)$ is constant. That is, the system is strictly conservative. Otherwise, there are losses (e.g., evaporation) in the system.

The flows in CNs can be defined according to different semantics [172]: (pure) *donor controlled*, when f_{ij} depends only on x_i ($f_{ij} = a_{ij} \cdot x_i$ in the linear case); (pure) *recipient controlled*, when f_{ij} depends only on x_j ($f_{ij} = b_{ij} \cdot x_j$ in the linear case); *donor and recipient controlled*, if f_{ij} depends on both, x_i and x_j (for example, $f_{ij} = c_{ij} \cdot x_i \cdot x_j$). Pure recipient controlled systems are *not* positive systems according to [62]. In an *unforced*, i.e., non-controlled, linear donor system, $\boldsymbol{A}$ is a Metzler matrix (non-diagonal elements are non-negative) and for any non-negative initial state the variables remain always non-negative, i.e., $\boldsymbol{x} \geq \mathbf{0}$ is a redundant constraint. As with QNs, the compartmental systems "view", has been considered for performance evaluation (transient and steady state), sensitivity and stability analysis or control design (for the last two see, see for example [74, 88]).

Compartmental and PN systems are considered in [154]. Due to the existence of synchronizations (joins or *rendez-vous*, and arc weights), it can be said that the structure of PNs is richer. Nevertheless, considering the semantics associated to the networks, in PNs there is a problem in representing recipient controlled compartmental systems in a "natural" way (because in PNs the flows are defined according to the marking of places at the precondition, not the marking of the subsequent places).

Using the same kind of "mass balance principle", in *Stochastic Flow Systems* (or *Stochastic Fluid Systems*) [30] both the *arrival* (or incoming) flow process and the *service* (or outgoing) process are random, normally assumed to be independent of each other. As in compartmental systems, $\boldsymbol{x}(t)$ represents the level of a set of reservoirs at time t, while controls may be applied to regulate the amount of flows. In this case, graphs illustrate flows, reservoirs and their connections. The key point is how equations are written, something for which there is a significant degree of freedom. In order to address optimization problems or sensitivity analysis, exogenous and endogenous events should be considered (the former referring to changes in the defining process, the latter characterizing points were the state of the system enters a certain region). In this framework, *Infinitesimal Perturbation Analysis* (IPA) has been successfully explored in several cases in order to optimize the behavior of the system (see, for example, [159, 177]).

As a third and last approach, let us mention *System Dynamics*. This is a modeling and analysis (basically bounded to simulation) methodology that began to be consolidated in parallel with Petri nets, in the 1960's. Jay W. Forrester started the System Dynamics Group at MIT, from which Systems Dynamics arose [64, 65]. Abstracting the possible discrete "nature" of the system under consideration, as in CNs or SFMs, systems are modeled as continuous or hybrid, now using two kinds of diagrams. Here we do not explicitly deal with the methodological aspects, but only to point out the existence of the so called *Forrester Diagrams* (FDs), also expressively known as *Stock and Flow Diagrams*. This kind of diagram allows the quantitative modeling of the relationships between the parts by means of a catalogue of symbols which correspond to a classical hydrodynamic interpretation of the system (see fig. 2).

The *stocks* correspond to the name of state variables in systems, while their values are the *levels* (stocks accumulate "material" coming from *material channels*); the *valves* determine the speed of the material flow through material channels (solid lines); the required information is transmitted instantaneously by means of *information channels* (dashed lines); *auxiliary variables* correspond to intermediate steps in the calculation of functions associated to the valves; the *clouds* represent sources and sinks; the interaction of the system with the exterior is represented by *exogenous* variables; the *delays* can affect the material of information transmission but they do not increase the modeling power of the formalism.

Elements of comparison between FDs and CPNs are provided in [90, 92]. At the structural level, as in EQNs, the variety of symbols in FDs contrasts with the frugality of basic symbols in PNs. Additionally, in FDs there exist not only *material flows*, but also graphical representations of *information flows*. Concerning the interpretation of the graph, as in compartmental systems (or SFMs, where noise is also considered) there is considerable freedom in defining the flows through arbitrary functions. In contrast, in CPNs the flow functions will be constrained by the particular server semantics (*finite, infinite, population* or *product*, etc.), using only local state variables in the precondition of the transition. Roughly speaking, if CNs, FDs and SFMs have increased expressivity in defining the "flows" quantitatively, the explicit presence of synchronizations in PNs makes the expression of simple tasks more natural as the assembly of two kinds of

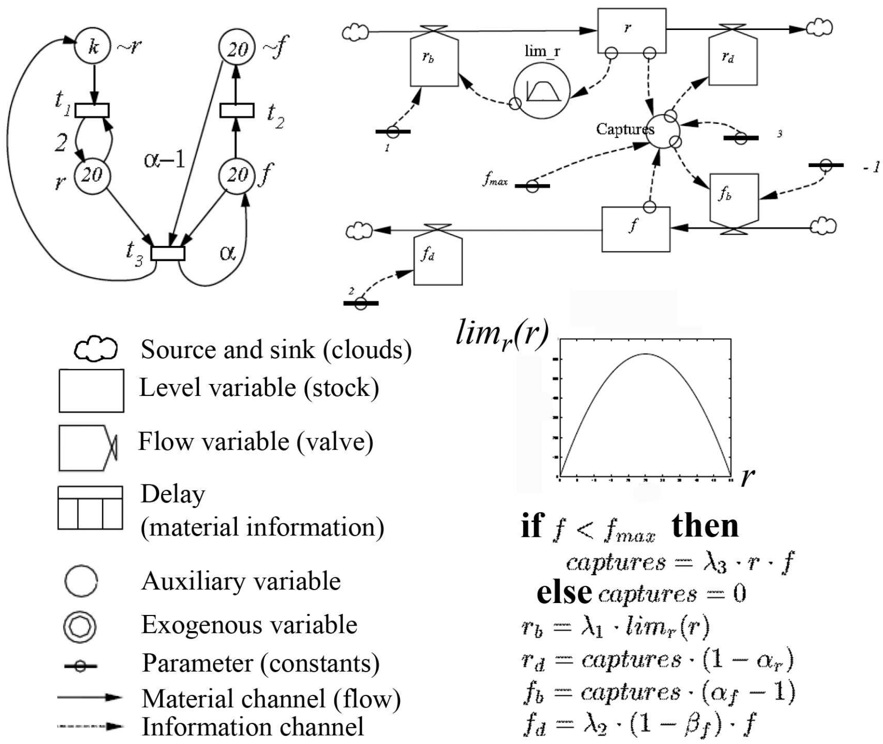


Figure 2: A continuous Petri net, the corresponding FD model and the basic components of a FD model. Observe that the simulation of the synchronization t_3 in the FD is rather "obscure".

parts.

To conclude, let us point out that (E)QNs, CNs, FDs or SFMs were directly defined as evolving in the time domain. Nevertheless, (C)PNs were primarily defined as fully non deterministic models, without any concept of time. This may be of particular interest in order to check some properties such as the potential presence of *deadlocks*, or the existence of other time independent constraints in behavior, e.g., certain *synchronic distances* or *mutual exclusions*.

However, the point here is not to focus on the differences among paradigms, but rather the reverse: in the end the different formalisms generate systems of equations that share some structural elements and there is a clear potential for transfering/adapting concepts and techniques.

2.3 Stochastic Process Algebras and fluid views

Returning to basic paradigms for modeling DEDS, let us consider *Process Algebras* (PAs), a formal paradigm created within the Computer Science community. If PNs were mainly defined to express "concurrency and synchronization", PAs base their basic modeling view on "components and composition" (let us say, a generalization of products of automata). Like PNs, PAs have been formally defined in a top-down manner, first without time, and in a second step adding the time dimension. Very recently, a lustrum ago, timed-fluid process algebras began to be explored.

From a historical perspective the three most classical process algebras are: *Calculus of Communicating Systems* (CSS) [127], *Communicating Sequential Processes* (CSP) [84], where local variables and the message passing paradigm of communication is adopted, and *Algebra of Communicating Processes* (ACP) [18], where the algebraic features are emphasized, introducing the noun phrase "process algebra". These proposals belong to an enormous set in which assumptions may sometimes differ very little, something which may puzzle outside observers of the field's evolution as in the case with QN and PN paradigms.

Work on *stochastic* PAs originated at the University of Erlangen at the beginning of the 1990s. Roughly speaking, they extend the untimed model with stochastic timings, and several proposals flourished during the first half of 1990s. They include [80]: TIPP (*TImed Processes and Performance evaluation*), PEPA (*Performance Evaluation Process Algebra*), MPA (*Markovian Process Algebra*), and EMPA (*Extended Markovian Process Algebra*).

Informally, it can be said that Petri Nets are to Stochastic Petri Nets (SPNs) what Process Algebras are to Stochastic Process Algebras (SPAs). SPNs were defined during the first half of the 1980s, while SPAs began to be defined one decade afterwards. Underlying both kinds of extensions are the goals of modeling and analysis of the functional behavior and performance characteristics of parallel and distributed DEDS. The goals in the first process algebras were defined so as to provide semantics for programming languages involving parallel constructions. This fact together with the textual programming style of defining models meant that the interest in process algebra was mostly confine to the Computer Science community.

If fluid or continuous PNs had a clear existence by the beginning of the 1990s [47, 48], fluid PAs began to be defined in the second half of the first decade of this century, i.e., about a decade and half later. Among these, PEPA [81] is a basic Markovian PA in which the basic components are sequential processes (finite automata), while parallel composition is only supported at the top level. Its fluid-flow approximation considers large scale models of massively repeated sequential components. The small set of combinators in PEPA contains *prefix* and *choice*, representing a sequential behavior and a choice, and *cooperation*, that defines synchronizations. Moreover, with *hiding* it is possible to abstract aspects of the components' behavior (roughly speaking, this is the "parallel" in PNs to *silent*, *immediate*, *non-observable*... transitions). Among other constraints on fluidization, components of the same type do not cooperate, i.e., synchronize. In a study related to chemical reactions, Cardelli (see, for example [28]) considers the fluidification of other SPAs. As with fluid QNs or continuous PNs, the basic goal is to look for a set of coupled *ordinary differential equations* (ODEs) as the underlying mathematical representation of the approximated behavior.

As in automata, in the PA paradigm the representation of the state is symbolic, a fact which is not appropriate for fluidization. As it is well-known, in PNs (particularly in *Place/Transition nets*, P/T nets), the distributed state is given by a numerical vector, the *marking*. Following this line of thinking, in [81] a numerically aggregated representation scheme is defined for PA expressions with replicated components. It explicitly introduces integer counters to define the state space, dealing with a state representation in *numerical vector form* which can be subject to a fluid-flow approximation. Obtained by aggregating all identical non-synchronized sequential components, the structure of the model is based on a so called *activity matrix* (its dimension being the number of *activities* by the number of distinct local *derivatives*), and timing is defined through a *rate function* per transition. Obviously the activity matrix is nothing more than the *inci-*

dence matrix of an underlying Petri Net (were activities are transitions and derivatives are places). This obvious fact has been recognized sometime later [54, 67]. Additionally, due to constraints on the subclass of considered process algebra models, the underlying Petri nets are *State Machine Decomposable*, an ordinary, i.e., no weights on the arcs, net subclass that is structurally bounded. Moreover, assuming interactions as being like those existing in computer networks, a *bounded capacity law* was considered in PEPA from the beginning. Components cannot perform activities any faster by cooperation, so the rate of a shared activity is the *minimum* of the apparent rates of the activity in the cooperating components. Therefore, the underlying CPN considers the so called *infinite server's semantics* for the fluid model (in subsequent works dealing with *signalling pathways*, borrowing terminology from chemistry, the *law of mass action* was added [39], replacing the minimum operator by the *product*, as in population dynamics [153]). A very positive point about this connection is that analysis and control of PEPA models can immediately use PN theory and techniques, and vice versa (a goal expressed in [57, 82], when comparing the expressiveness of PEPA and bounded SPN models). Extensions of this subclass of PA models can be found in [76] where more than constructing a limiting-deterministic approximation, higher order moments (in particular, variances) are estimated. This is necessary for guessing the accuracy of the fluid-flow approximation in a given situation.

2.4 Some remarks on the sketched landscape

The present section makes a long trip flying over the big forest (do not translate this last word into Latin, to avoid certain confusions) of fluid models of discrete event dynamic systems. As a summary of some of the ideas:

- Starting from DEDS views based on "customers-services" relationships (QNs), expressing "concurrency and synchronization" (PNs), or "components and composition" (PAs), the fluid models have similar kinds of structure, with meaningful graphical representation.
- In essence, most of the considered DEDS models and their fluid relaxations directly reflect a *bipartite* structure: queues in QNs, places in PNs or storages (tanks) in FDs are "containers", while stations in QNs, transitions in PNs or valves in FDs deal with "activities" (in the QN case by agents identified as servers). Roughly speaking, this represents a *consumer/production* logic, a feature needed for manufacturing, communications, logistics, distributed computations, transportation (road traffic), chemical-reactions, biological or ecological systems, to give some examples. The distributed state is (partially) represented by the number of customers in QNs, tokens in PNs or levels in FDs in the model.
- The presence of synchronizations (joins and arc-weights in PNs) and very different stochastic interpretations (where the QNs literature is richer) are the more significant peculiarities of the particular kind of equations in fluid models.
- In PNs, the state is always considered in a *distributed* and *numerical* way, as opposite to the central and symbolic view (single global state-variable) provided by *automata* or *Markov chains*. This is also the reason why fluid PA models pass through an intermediate PN representation.
- The accuracy of the approximation depends on the structure of the model, the timing, the initial state and the performance metrics of interest.

- Under appropriate stability conditions, the classical robustness of closed-loop control, i.e., reduction of sensitiveness, can more easily exploit the fluid approximation than those required for pure performance evaluation problems.

- The considered fluid models ("approximate or not") are technically (peculiar) *hybrid systems*, due to upper/lower bounds on actions and states, or to flow definitions as those in which a *minimum* operator is employed (as in the so called *infinite server semantics* in continuous PNs).

- In many cases, in sound theories to deal with fluid approximation of large scale DEDS, it would be important to have not only a timed approximation, but also an untimed one, i.e., assuming full non-determinism, where pure logical properties of the system can be studied (such as deadlock freeness or liveness, structural boundedness-stability, synchronic properties...).

Once models of QNs, PNs or PAs are fluidified, or CSs, SFMs or FDs are considered, the kind of (stochastic) equations that can be obtained have many structural peculiarities in common. In other words, the study of those fluid models may apply to broader frameworks than the precise DEDS modeling paradigm from which they originate. Perhaps one way of understanding this important fact is as follows: if all the performance models use exponential probability density functions (pdfs), the lower level models reduce to Markov Chains, and the mentioned proximity among fluid approximations can be "easily accepted". If the pdfs are non exponential, under high utilization probability of servers (heavy traffic...), functional central limit theorems will "uniform" the stochastic kind of fluid models; for really very large populations, deterministic differential equations may be truly appropriate. In some sense, an idea of this kind is expressed in [75], where a global reflection on the proper mathematical setting for systems of the type being considered is explained. Presented from a management/operations research perspective as an extension of classical linear programming models (static and deterministic, appropriated for very long terms), Harrison introduces time dimension and random behaviors, identifying the existence of *resources, buffers, activities* and "materials" (units of flow). The formalism introduced is called *Stochastic Processing Networks*, its roots lying in the classical *Activity Analysis*, began in the 1950s.

The quantity of works on fluid QNs today is impressive. At the other extreme, the more recent approach to fluidization of DEDS concerns PAs, which goes through a numerical PN-based representation. In a very simplistic way, the communities of QNs, and following that lines, considering in fact a subclass of PNs, the community of PAs pay important attention to the justification of the fluid models using the functional law of large numbers and functional central limit theorems. Most frequently the results concern populations growing to infinity, while time is kept finite. Alternatively, in the context of PNs the consideration of steady-state (time going to infinite) for big (but bounded) populations has been studied (see sec. 4.6). For QNs there is a great abundance of studies concerning stability or optimization (parametric and dynamic) issues. In PNs the extensive use of structure theory (see, for example [158]) to deal with functional properties for the underlying non-deterministic discrete model is a peculiar feature, while some bridges to automatic control concepts and techniques have been explored (see, for example, hereafter the sections 4 and 5).

3 Fluidification of untimed net models

After the previous broad perspective, let us now concentrate on Petri nets. This section presents the formalism of continuous Petri nets and its behavior in the untimed framework. It deals with basic concepts, as $\lim$-reachability and desired logical properties, and relates them to those ones of the discrete systems.

3.1 Basic concepts and definitions

In the following, it is assumed that the reader is familiar with Petri nets (PNs) (see [49, 132, 149] for an introduction). The usual PN system will be denoted as $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$, where $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is the net structure:

- P and T are disjoint and finite sets of places and transitions;
- $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ are $|P| \times |T|$ sized, natural valued, incidence matrices. The net is said to be *ordinary* if $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ are valued on $\{0, 1\}$;

and $\boldsymbol{M}_0 \in \mathbb{N}_{\geq 0}^{|P|}$ is the initial (discrete) marking. A continuous system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is understood as the fluid relaxation of all the transitions of a *discrete* system. The main difference between continuous and discrete PNs is in the firing count vector and consequently in the marking, which in discrete PNs are restricted to natural numbers, while in continuous PNs are relaxed to non-negative real numbers, e.g., $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$. Observe that uppercase $\boldsymbol{M}$ represents the marking of a discrete net system, while lowercase $\boldsymbol{m}$ represents the marking of a continuous net system. In the following, it will be assumed that all the components of the firing count vector are non-negative real numbers, what implies a full relaxation of the system. The marking of a place of a continuous system can be seen as an amount of fluid stored in the place, and the firing of a transition can be considered as a flow of fluids going from the set of its input places to the set of its output places (in general, fluids can be created or destroyed by firing a transition, because any local conservation of material is required).

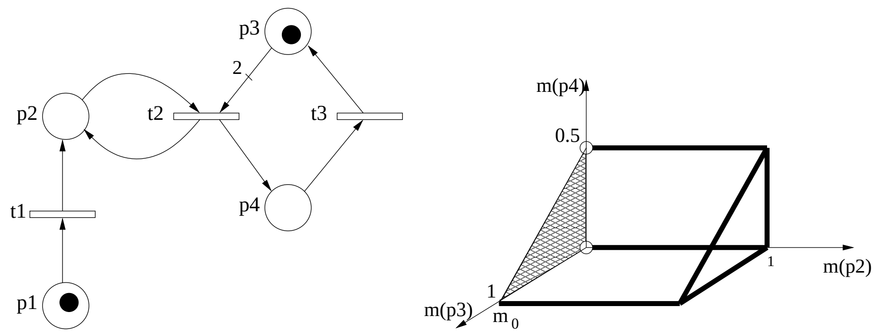


Figure 3: (a) Autonomous continuous system (b) Lim-Reachability space

Given a node $v \in P \cup T$, its *preset*, $^\bullet v$, is defined as the set of its input nodes, and its *postset*, $v^\bullet$, as the set of its output nodes. For example, in the PN of fig. 3(a), $^\bullet t_2 = \{p_2, p_3\}$, while $p_3{}^\bullet = \{t_2\}$. These definitions can be naturally extended to sets of nodes. A transition t is *enabled* at $\boldsymbol{m}$ if for every $p \in {}^\bullet t$, $\boldsymbol{m}[p] > 0$. In other words, the enabling condition of continuous systems and that of discrete ordinary systems can be expressed in an "analogous" way: every input place should be marked. 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 degree* of a transition which, for continuous nets, is defined for a given marking $\boldsymbol{m}$ as

$$\text{enab}(t, \boldsymbol{m}) = \min_{p \in {}^\bullet t} \frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p, t]} \tag{1}$$

The enabling degree of a transition represents the maximal amount in which the transition can be fired in a single occurrence. In this section no policy for the firing of transitions is imposed, that is, a full non-determinism is assumed for the order and firing amounts in which transitions are fired.

The firing of t in a certain amount α, with $0 < \alpha \leq \text{enab}(t, \boldsymbol{m})$ leads to a new marking $\boldsymbol{m}'$, and it is denoted as $\boldsymbol{m} \xrightarrow{\alpha t} \boldsymbol{m}'$. It holds $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[P, t]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow* matrix (*incidence* matrix if $\mathcal{N}$ is self-loop free), and $\boldsymbol{C}[P, t]$ is the column of $\boldsymbol{C}$ devoted to transition t. Hence, as in discrete systems, the state (or fundamental) equation

$$\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \;\; \boldsymbol{m}, \boldsymbol{\sigma} \geq 0 \tag{2}$$

summarizes the way the marking evolves. As it will be discussed, for discrete models the state equation $\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \;\; \boldsymbol{M}, \boldsymbol{\sigma} \geq 0$ provides a necessary condition for a marking to be reachable, however it is not a sufficient condition since it can contain *spurious* solutions, i.e., non reachable solutions.

The *support* of a vector $\boldsymbol{v} \geq \boldsymbol{0}$ is $\|\boldsymbol{v}\| = \{v_i | v_i > 0\}$, the set of positive elements of $\boldsymbol{v}$. Right and left natural annullers of the token flow matrix are called T- and P-*semiflows*, respectively. A semiflow is *minimal* when its support is not a proper superset of the support of any other semiflow, and the greatest common divisor of its elements is one. As in discrete nets, when $\boldsymbol{y}^T \cdot \boldsymbol{C} = \boldsymbol{0}$, $\boldsymbol{y} > \boldsymbol{0}$ the net is said to be *conservative*, and when $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{x} > \boldsymbol{0}$ the net is said to be *consistent*.

P-semiflows lead to three different concepts: a) the P-semiflow itself which is a non-negative vector ($\boldsymbol{y} \geq \boldsymbol{0}$, $\boldsymbol{y}^T \cdot \boldsymbol{C} = \boldsymbol{0}$); b) the conservation law induced by the P-semiflow, i.e., if $\exists \boldsymbol{y} \gneq \boldsymbol{0}$ then, by the state equation, it holds that given an arbitrary $\boldsymbol{m}_0$, $\boldsymbol{y}^T \cdot \boldsymbol{m}_0 = \boldsymbol{y}^T \cdot \boldsymbol{m}$ for every reachable marking $\boldsymbol{m}$; c) the subnet generated by the places in the support of the P-semiflow ($P_y = \|\boldsymbol{y}\|$, $T_y = {}^\bullet P_y \cup P_y{}^\bullet$), a *P-conservative component*. On the other hand, T-semiflows also admit three views: a) the non-negative vector that is a right annuller of the incidence matrix ($\boldsymbol{x} \geq \boldsymbol{0}$, $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$); b) the potentially cyclic behaviours induced by the T-semiflow. i.e., if $\exists \boldsymbol{x} \gneq \boldsymbol{0}$ that is fireable from $\boldsymbol{m}_0$ then, by the state equation, $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}_0$ with σ being a firing sequence whose firing count vector equals $\boldsymbol{x}$; c) the subnet generated by the transitions in the support of the T-semiflow ($T_x = \|\boldsymbol{x}\|$, $P_x = {}^\bullet T_x \cup T_x{}^\bullet$).

For example, the PN in fig. 1 has several P-semiflows and one of them is a vector $\boldsymbol{y}$ with zero elements except $y_9 = y_{13} = 1$, i.e., $\|\boldsymbol{y}\| = \{p_9, p_{13}\}$. Observe that, from the initial marking in the figure, $\boldsymbol{m}[p_9] + \boldsymbol{m}[p_{13}] = 1$ for any reachable marking $\boldsymbol{m}$. Alternatively, observe that one of the T-semiflows is the vector with all elements equal to 1 excepting components 3 and 5 that are equal to 0. Thus, if all transitions are fired excepting t_3 and t_5, the final marking that is reached is the same as the initial one. Therefore, the PN has a T-semiflow corresponding to this sequence $\boldsymbol{x}$ with all elements equal to 1 excepting $x_3 = x_5 = 0$.

A set of places Σ is a *siphon* if ${}^\bullet\Sigma \subseteq \Sigma^\bullet$. A set of places Θ is a *trap* if: (a) $\Theta^\bullet \subseteq {}^\bullet\Theta$; and (b) for each place $p \in \Theta$ the firing of any $t \in {}^\bullet p$ enables at least one $t \in p^\bullet$. Condition (b) is always satisfied in CPNs and in ordinary discrete PNs. For non-ordinary discrete PNs, condition (b) is satisfied if a non-blocking condition is true [22][3]. Therefore, for ordinary nets or if the non-blocking condition is ignored, a trap in $\mathcal{N}$ is a siphon in the reverse net $\mathcal{N}^r$, i.e., the resulting net of reversing all arcs. In discrete nets, initially marked traps cannot be emptied. More formally, let $\Theta = \|\boldsymbol{y}\|$ be a trap, if $\boldsymbol{y}^T \cdot \boldsymbol{M}_0 \geq 1$ then $\boldsymbol{y}^T \cdot \boldsymbol{M} \geq 1$ for any reachable marking $\boldsymbol{M}$. Symmetrically, initially empty siphons cannot get marked, i.e., let $\Sigma = \|\boldsymbol{y}\|$ be a siphon, if $\boldsymbol{y}^T \cdot \boldsymbol{M}_0 = 0$ then $\boldsymbol{y}^T \cdot \boldsymbol{M} = 0$ for any reachable marking $\boldsymbol{M}$.

For the same PN in fig. 1, $\Theta = \{p_5, p_7\}$ is a trap since $\Theta^\bullet = \{t_5\} \subseteq \{t_3, t_5\} = {}^\bullet\Theta$, while $\Sigma = \{p_9, p_{10}, p_{13}\}$ is a siphon since: ${}^\bullet\Sigma = \{t_9, t_{10}\} \subseteq \{t_9, t_7, t_{10}\} = \Sigma^\bullet$.

The definitions of subclasses that depend only on the structure of the net are also generalized to continuous nets. For instance, in *choice free* nets (CF) each place has at most one output transition. In *equal conflict* nets (EQ) all conflicts are equal, i.e., ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \boldsymbol{Pre}[P,t] = \boldsymbol{Pre}[P,t']$ (for instance transitions t_3 and t_4 in fig. 1 are in equal conflict). Moreover, a net $\mathcal{N}$ is said to be *proportional equal conflict* if ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \exists q \in \mathbb{R}_{>0}$ such that $\boldsymbol{Pre}[P,t] = q \cdot \boldsymbol{Pre}[P,t']$. A net $\mathcal{N}$ is said to be *mono-T-semiflow* (MTS) if it is conservative and has a unique minimal T-semiflow whose support contains all the transitions.

3.2 Fireable sequences, reachability sets and a necessary condition for fluidization

In order to illustrate the firing rule in a continuous system, let us consider the system in fig. 3(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, firing by 0.5 would yield marking $\boldsymbol{m}_1 = [0.5\ 0.5\ 1\ 0]^T$. At $\boldsymbol{m}_1$ transition t_2 has enabling degree equal to 0.5; if it is fired in this amount the resulting marking is $\boldsymbol{m}_2 = [0.5\ 0.5\ 0\ 0.5]^T$. Both $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are markings reachable with finite firing sequences, or simply reachable markings.

For a given system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, the set of all markings that are reachable by a finite number of firings is denoted as $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$. Interestingly this set is *convex* [142].

Proposition 1 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a continuous PN system. The set $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ is convex, i.e., if two markings $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are reachable, then for any $\alpha \in [0,1]$, $\alpha\boldsymbol{m}_1 + (1-\alpha)\boldsymbol{m}_2$ 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 (equivalently, there exists no empty, i.e., unmarked, siphon). From this, realizability of T-semiflows can be deduced [142], and therefore behavioral and structural *synchronic relations* [148,150] coincide in consistent continuous systems in which every transition is fireable at least once. In particular, defining boundedness and structural boundedness as in discrete systems (a system is *bounded* iff $k \in \mathbb{N}$ exists such that for every

[3]For each place in the trap, the minimum weight of the input arcs is greater than or equal to the minimum weight of its output arcs, i.e., $\forall p \in \Theta$ such that ${}^\bullet p \neq \emptyset$ it holds that $min_{t_i \in {}^\bullet p} \boldsymbol{Post}[p, t_i] \geq min_{t_o \in p^\bullet} \boldsymbol{Pre}[p, t_o]$.

reachable marking $\boldsymbol{m} \leq k \cdot \mathbf{1}$, and it is *structurally bounded* iff it is bounded with every initial marking), it is immediate to see that both concepts coincide in continuous systems in which every transition is fireable. And, as in discrete systems, structural boundedness is equivalent to the existence of $\boldsymbol{y} > \mathbf{0}$ such that $\boldsymbol{y} \cdot \boldsymbol{C} \leq \mathbf{0}$ (see, for example, [22, 158]).

Assume that the initial marking of a given system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is a vector of non-negative integers, i.e., $\boldsymbol{m}_0 \in \mathbb{N}_{\geq 0}^{|P|}$. Obviously, if $\boldsymbol{m}$ is a marking that is reached by firing transitions in discrete amounts, i.e., as if the system was discrete, then $\boldsymbol{m}$ is also reachable by the system as continuous just by applying the same firing sequence. Thus $\mathrm{RS}_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq \mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ where $\mathrm{RS}_D(\mathcal{N}, \boldsymbol{m}_0)$ is the discrete reachability set, i.e., the set of markings reachable by the system as discrete. An immediate consequence of this is that boundedness of the continuous system is a sufficient condition for boundedness of the discrete system.

A marking is said to be lim-reachable if it can be reached with a possibly infinite firing sequence. More formally:

Definition 2 *[142] Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a continuous system. A marking $\boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|}$ is* lim-reachable*, if a sequence of reachable markings $\{\boldsymbol{m}_i\}_{i \geq 1}$ exists such that*

$$\boldsymbol{m}_0 \xrightarrow{\sigma_1} \boldsymbol{m}_1 \xrightarrow{\sigma_2} \boldsymbol{m}_2 \cdots \boldsymbol{m}_{i-1} \xrightarrow{\sigma_i} \boldsymbol{m}_i \cdots$$

and $\lim_{i \to \infty} \boldsymbol{m}_i = \boldsymbol{m}$.

The lim-reachable space is the set of lim-reachable markings, and will be denoted lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$. Fig. 3(b) depicts lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ of the system in fig. 3(a). It is not necessary to represent the marking of place p_1 since $\boldsymbol{m}[p_1] = 1 - \boldsymbol{m}[p_2]$ (p_1 and p_2 define a token conservation law). The set of lim-reachable markings is composed of the points inside the prism, i.e., the interior points, the points in the non shadowed sides, the points in the thick edges and the points in the non circled vertices.

Let us consider again the system in fig. 3(a) with initial marking $\boldsymbol{m}_0 = [0.5\ 0.5\ 0\ 0.5]^T$. The firing of t_3 in an amount of 0.5 makes the system evolve to marking $[0.5\ 0.5\ 0.5\ 0]^T$ from which t_2 can be fired in an amount of 0.25 leading to marking $[0.5\ 0.5\ 0\ 0.25]^T$. Now, the markings of places p_1, p_2 and p_3 are the same as those of the system at $\boldsymbol{m}_0$, but the marking of p_4 is half of its marking at $\boldsymbol{m}_0$. As transitions t_2 and t_3 are further fired, the marking of p_4 approaches 0. Notice that the marking reached in the limit $[0.5\ 0.5\ 0\ 0]^T$ corresponds to the emptying of an initially marked trap $\Theta = \{p_3, p_4\}$, fact that can not occur in discrete systems. Thus, in continuous systems traps may not *trap*! From the point of view of the analysis of the behaviour of the system, it is interesting to consider this lim-reachable marking, since it is the one to which the state of the system may converge.

For any continuous system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, the differences between $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ and lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ are just in the border points of their convex spaces. In fact, it holds that $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0) \subseteq$ lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ and that the closure of $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$, i.e., all the points in $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ plus the limit points of $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$, is equal to the closure of lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ [99].

As in discrete systems, a continuous system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is said to *deadlock* if a marking $\boldsymbol{m} \in \mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ exists such that $\mathrm{enab}(t, \boldsymbol{m}) = 0$ for every transition t; the system is *live* if for every transition t and for any marking $\boldsymbol{m} \in \mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ a successor $\boldsymbol{m}'$ exists such that $\mathrm{enab}(t, \boldsymbol{m}') > 0$; and a net $\mathcal{N}$ is *structurally live* if $\exists\, \boldsymbol{m}_0$ such that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is live.

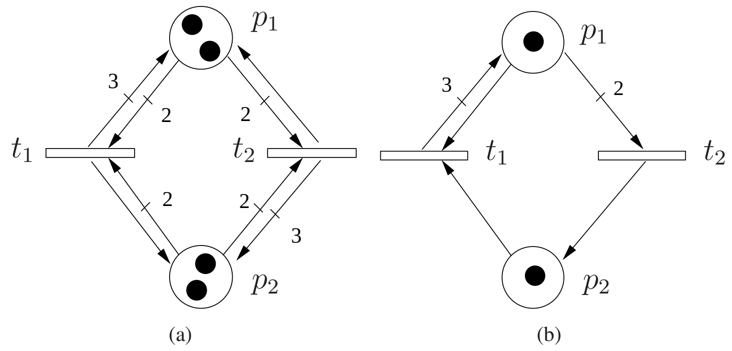


Figure 4: Two MTS systems that behave in very different ways if seen as discrete or as continuous.

The fact that $\text{RS}_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq \text{RS}(\mathcal{N}, \boldsymbol{m}_0)$ might involve the loss of some properties of the discrete system, e.g., the new reachable markings might make the system live or might deadlock it. The system in fig. 4(a) deadlocks as discrete after the firing of transition t_1. However, it never gets completely blocked as continuous unless an infinitely long sequence is considered. On the other hand, the system in fig. 4(b) is live as discrete but gets blocked as continuous if transition t_2 is fired in an amount of 0.5. This *non-fluidizability* of discrete net systems with respect to the deadlock-freeness property (also with respect to liveness because they are MTS nets), 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 (for example, due to the existence of a chaotic behavior).

It must be pointed out that a system can be fluidizable with respect to a given property, i.e., the continuous model preserves that property of the discrete one, but not with respect to other properties. Thus, the usefulness of continuous relaxations of discrete models depends not only the systems being studied but also on the properties to be analyzed.

Interestingly, the set $\text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$ can be easily characterized if some common conditions that can be checked in polynomial time are fulfilled [142].

Proposition 3 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be consistent and such that each transition can be fired at least once. Then $\boldsymbol{m} \in \text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$ iff there exists $\boldsymbol{\sigma} > 0$ such that $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$.*

Hence, if a net is consistent and the system has no empty siphon at $\boldsymbol{m}_0$, then the set of lim-reachable markings is fully characterized by the state equation. This immediately implies *convexity* of $\text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$ and the inclusion of every spurious discrete solution in $\text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$. Recall that $\boldsymbol{m}$ is said to be a *spurious* discrete solution if $\boldsymbol{m}$ is solution of the state equation, i.e., there exists $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}_{\geq 0}$ such that $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, but $\boldsymbol{m}$ is not reachable, i.e., $\boldsymbol{m} \notin \text{RS}_D(\mathcal{N}, \boldsymbol{m}_0)$. Fortunately, as it will be shown in the next section, every spurious solution in the border of the convex set $\text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$ can be *cut* by adding some implicit places (more precisely the so-called *cutting implicit places* [42]) what implies clear improvements in the state equation representation. Improvements in the computation of performance bounds for discrete PNs is considered in [26].

If $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is not consistent or some transitions cannot be fired, $\text{lim-RS}(\mathcal{N}, \boldsymbol{m}_0)$ can still be characterized by using the state equation plus a simple additional constraint concerning the fireability of the transitions in $\|\boldsymbol{\sigma}\|$. The set $\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ can also be fully determined by adding one further constraint related to the fact that a finite firing sequence cannot empty a trap [99] (in contrast to infinite sequences which might empty initially marked traps as shown in this section).

3.3 Liveness conditions for continuous systems

Liveness and deadlock definitions can be straightforwardly extended for the concept of lim-reachability.

Definition 4 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a continuous PN system.*

- $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ lim-deadlocks *if a marking $\boldsymbol{m} \in$ lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ exists such that* $\mathrm{enab}(t, \boldsymbol{m}) = 0$ *for every transition t;*
- $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *is* lim-live *iff for every transition t and for any marking $\boldsymbol{m} \in$ lim-$\mathrm{RS}(\mathcal{N}, \boldsymbol{m}_0)$ a successor $\boldsymbol{m}'$ exists such that* $\mathrm{enab}(t, \boldsymbol{m}') > 0$*;*
- $\mathcal{N}$ *is* structurally lim-live *if $\exists\, \boldsymbol{m}_0$ such that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is lim-live.*

Notice that although lim-*deadlocks* may only be reached in the limit, they highlight an important system weakness: they allow the system to reach a marking in which all transitions have either 0 or infinitely small enabling degrees.

As discussed in the previous subsection, the state equation provides a full characterization of the lim-reachable markings for consistent nets with no empty siphons. This allows one to use the state equation to look at $\boldsymbol{m}_0$ for deadlocks, i.e., markings at which every transition has at least one empty input place. Consider the net in fig. 5 with $\boldsymbol{m}_0 = [10\ 11\ 0]^T$. It is consistent (with $\boldsymbol{x}_1 = [1\ 1]^T$ as its only minimal T-semiflow) and conservative (with $\boldsymbol{y}_1 = [1\ 0\ 1]^T$ and $\boldsymbol{y}_2 = [0\ 1\ 1]^T$ as minimal P-semiflows). At any potential lim-deadlock marking $\boldsymbol{m}$, both transitions t_1 and t_2 must be disabled, i.e., at least one input place per transition is empty. Thus, transition t_1 is disabled iff $\boldsymbol{m}[p_1] = 0$ or $\boldsymbol{m}[p_2] = 0$, and transition t_2 is disabled iff $\boldsymbol{m}[p_1] = 0$ or $\boldsymbol{m}[p_3] = 0$. Hence, at a lim-deadlock marking $\boldsymbol{m}$ it holds $\boldsymbol{m}[p_1] = 0 \vee (\boldsymbol{m}[p_2] = 0 \wedge \boldsymbol{m}[p_3] = 0)$. As stated, this problem might be directly associated to a satisfiability problem, which has exponential complexity. Alternatively, deadlock-freeness can be straightforwardly expressed as a set of non-linear (bi-linear) equations. Let us define $\mathbf{Pre}_\Sigma$ and $\mathbf{Post}_\Sigma$ as $|P| \times |T|$ sized matrices such that:

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

Equations $\{\boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \le \mathbf{0},\ \boldsymbol{y} \ge \mathbf{0}\}$ where $\boldsymbol{C}_\Sigma = \mathbf{Post}_\Sigma - \mathbf{Pre}_\Sigma$ define a generator of siphons (Σ is a siphon iff $\exists \boldsymbol{y} \ge \mathbf{0}$ such that $\Theta = \|\boldsymbol{y}\|$, $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \le \mathbf{0}$) [61, 158].

Proposition 5 *The following system:*

- $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \ \ \boldsymbol{m}, \boldsymbol{\sigma} \ge 0,$ *{state equation}*
- $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \le \mathbf{0}, \boldsymbol{y} \ge \mathbf{0},$ *{siphon generator}*
- $\boldsymbol{y}^T \cdot \boldsymbol{m} = 0,$ *{empty siphon at $\boldsymbol{m}$}*
- $\boldsymbol{y}^T \cdot \mathbf{Pre} \ge \mathbf{1},$ *{at least one input place per transition}*

has no solution iff the continuous net system is deadlock-free.

Proposition 5 is derived from the statements that correspond to each constraint of the linear system. The existence of a reachable marking, in which a siphon that contains at least one input place per transition is empty, is a necessary and sufficient condition for non-deadlock-freeness. Notice that if the last constraint $\boldsymbol{y}^T \cdot \mathbf{Pre} \geq \mathbf{1}$ is removed, then activity in some transitions is allowed, and hence the existence of solution for the remaining constraints represent a necessary and sufficient condition for non-liveness.

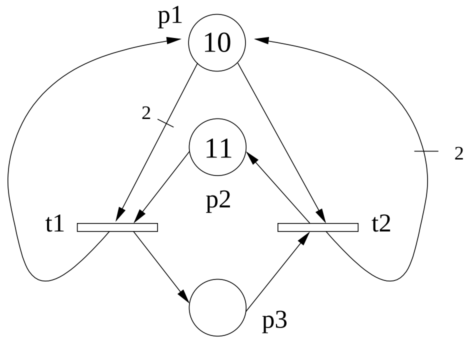


Figure 5: A continuous MTS system that integrates a discrete spurious deadlock $\boldsymbol{m} = [0\ 1\ 10]^T$, reachable through the firing sequence $5t_1, 2.5t_1, 1.25t_1, \ldots$.

The set of places $\{p_2, p_3\}$ in fig. 5 is the support of an initially marked P-semiflow, and therefore both places cannot be emptied simultaneously. This implies that a deadlock occurs iff p_1 is emptied. The marking $\boldsymbol{m} = [0\ 1\ 10]^T$ can be obtained as a solution of the state equation with $\boldsymbol{\sigma} = [10\ \ 0]^T$ as firing count vector. Thus given that the system satisfies the conditions of Proposition 3, $\boldsymbol{m}$ is $\lim$-reachable, i.e., the continuous 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. However, it is well known that initially marked traps cannot be completely emptied in discrete nets. Thus, $\boldsymbol{m}$ is a spurious solution of the state equation if we consider the system as discrete. An important question is now: How to search for and to remove (discrete) spurious solutions, i.e., non-reachable markings?

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 $\{\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0},\ \boldsymbol{y} \geq \mathbf{0}\}$ where $\boldsymbol{C}_\Theta = \mathbf{Post}_\Theta - \mathbf{Pre}_\Theta$ define a generator of traps (Θ is a trap iff $\exists \boldsymbol{y} \geq \mathbf{0}$ such that $\Theta = \|\boldsymbol{y}\|$, $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0}$) [61, 158]. Hence, given $\boldsymbol{m}$ we can check in polynomial time a sufficient condition for a solution of the state equation to be spurious:

Proposition 6 *Given* $\boldsymbol{m} \in \mathbb{N}_{\geq 0}^{|P|}$ $(\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \ \boldsymbol{m}, \boldsymbol{\sigma} \geq 0)$*, if*

- $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0}, \boldsymbol{y} \geq \mathbf{0}$, $\{$*trap generator*$\}$
- $\boldsymbol{y}^T \cdot \boldsymbol{m}_0 \geq 1$, $\{$*initially marked trap*$\}$
- $\boldsymbol{y}^T \cdot \boldsymbol{m} = 0$, $\{$*trap empty at* $\boldsymbol{m}\}$

has solution, then $\boldsymbol{m}$ *is a discrete spurious solution.*

The result of Proposition 6 follows directly from the fact that $\|\boldsymbol{y}\|$ is a trap that has been emptied. Fortunately, there exist techniques to *cut* spurious solutions of the state equation [42]. Let us show how the spurious solution $\boldsymbol{m} = [0\ 1\ 10]^T$ can be cut by adding a place that in discrete net is implicit. Recall that 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.

Since p_1 is an initially marked trap, its marking must satisfy $\boldsymbol{m}[p_1] \geq 1$. This equation together with the conservation law $\boldsymbol{m}[p_1] + \boldsymbol{m}[p_3] = 10$ leads to $\boldsymbol{m}[p_3] \leq 9$. This last inequality can be forced by adding a slack variable, i.e., a *cutting implicit place* q_3, such that $\boldsymbol{m}[p_3] + \boldsymbol{m}[q_3] = 9$. 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 with higher marking) and therefore p_2 can be removed without affecting the system behavior. In the resulting net system, $\boldsymbol{m} = [0\ 1\ 10]^T$ 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.

Notice that in continuous systems, deadlock markings are always in the borders of the convex set of reachable markings and hence, discrete spurious deadlocks can be cut by the described procedure. This way, the addition of cutting implicit places improves the quality of the continuous net as an approximation of the discrete one by eventually increasing the number of P-semiflows and traps. Notice that such an addition creates more traps that might be treated similarly (if they are the cause of spurious solutions) in order to improve further the quality of the continuous approximation.

It must be pointed out that the concept of limit-reachability in continuous nets provides an interesting approximation to discrete nets in the sense that $\lim$-liveness of the continuous system is a sufficient condition for liveness of the discrete one [142]:

Proposition 7 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a bounded and lim-live system. Then, $\mathcal{N}$ is structurally live and structurally bounded as a discrete net.*

From Proposition 7 it is clear that any necessary condition for a discrete system to be structurally live and structurally bounded, is also necessary for it to be structurally lim-live and bounded. In particular rank theorems [141] 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. These are equivalence relations, and the sets of all the equal conflict and proportional equal conflict sets are denoted by SEQS (e.g., the set $\{t_3, t_4\}$ in fig. 1 is an equal conflict set) and SPEQS (e.g., the set $\{t_3, t_4\}$ in fig. 1 is a proportional equal conflict set for any weights of the arcs connecting p_4 to t_3 and t_4) respectively. The following rank theorem [142] establishes a necessary condition for $\lim$-liveness:

Proposition 8 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a bounded and lim-live system. Then, $\mathcal{N}$ is consistent, conservative and $rank(\boldsymbol{C}) < |SPEQS|$.*

In discrete EQ systems another rank theorem provides a full characterization of structural liveness and structural boundedness [161]. For continuous EQ systems this result can be extended leading to a full characterization of lim-liveness and boundedness of polynomial time complexity [142]:

Proposition 9 *A continuous EQ system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is lim-live and bounded iff:*

- *$\mathcal{N}$ is consistent, conservative, $rank(\boldsymbol{C}) = |SPEQS| - 1$ and*

Clients	Servers	Semantics of the transition
Many	Many	infinite server semantics
Many	Few	finite server semantics
Few	Few	discrete transitions
Few	Many	discrete transitions

Table 1: A qualitative approach to the fluidization of a transition [155]

- *The support of every P-semiflow is marked (* $\nexists \boldsymbol{y}^T \geq 0, \boldsymbol{y}^T \cdot \boldsymbol{C} = \boldsymbol{0}, \boldsymbol{y} \cdot \boldsymbol{m}_0 = \boldsymbol{0}$*).*

Let us finally notice that there exist transformation techniques, namely *equalization* and *release*, that convert non EQ systems into EQ ones and, under some conditions, preserve non (structural) liveness. While equalization *hardens* the enabling conditions of the transitions to make them equal, release *weakens* such conditions. Thus, these transformations allow to obtain some sufficient liveness conditions for non EQ systems out of the ones known for EQ systems (see [141] for details).

4 Fluidization of Timed net models

This section introduces the notion of time in the continuos Petri net formalism presenting the most used firing semantics. Then, the main focus will be on *infinite server semantics*. Some basic properties such that the monotonicity of steady-state throughput or the relation of liveness with untimed model is considered. Even if some properties are undecidable, a model checking technique is also mentioned. Finally, some considerations on how can be improved the approximation are done considering the removing of spurious solutions, the addition of noise, or the definition of *ad-hoc* server semantics.

4.1 Conceptual framework and server semantics

If a timed interpretation is included in the *continuous* model, the fundamental equation explicitly depends on time: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$, which, through time differentiation, becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing sequence $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$ is called the *(firing) flow*, and leads to the following equation for the dynamics of the timed CPN (TCPN) system:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{f}(\tau). \tag{3}$$

Depending on how the flow $\boldsymbol{f}$ is defined, different firing semantics can be obtained. In general, transitions are interpreted as *stations* (in QN terminology), where *servers* and *clients* meet. Thus, “a priori” the most appropriate firing relaxation depends on the relative number of servers and clients in the discrete model that we want to approximate. Very roughly speaking, assuming that there may be “many” or “few” of each of them, fluidization can be considered for clients, for servers or for both. Table 4.1 represents qualitatively the four possible cases. If the number of clients is “small” (Few-Few and Few-Many in Table 4.1), the system is not too much “crowded”, the transitions “should” remain discrete and the fluidization may be unsuitable. If there are many clients and few servers (Many-Few) the relaxation is only at the level of clients, and the so called *finite server semantics* may provide a good approximation. On the other hand, in the case of many clients and many servers (Many-Many), a continuous model with the so called *infinite server semantics* seems reasonable, since there are so many servers that there is no need to make them explicit.

Let us assume that a constant λ_j is assigned to each transition t_j. For *finite server semantics*, 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 empty places (hence, λ_j represents the product of the number of servers in the transition and their speed). This corresponds to the *constant speed* of [2], where the flow of a transitions t_j is:

$$f_j = \begin{cases} \lambda_j, \text{ if } \forall p_i \in {}^\bullet t_j, m_i > 0 \\ \min\left\{ \min\limits_{p_i \in {}^\bullet t_j | m_i = 0} \left\{ \sum\limits_{t_q \in {}^\bullet p_i} \frac{f_q \cdot \boldsymbol{Post}[t_q, p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}, \lambda_j \right\}, & \text{otherwise} \end{cases} \quad (4)$$

The dynamical system corresponds to a *piecewise constant* system; a switch occurs when the set of empty places changes and the new flow values must ensure that the marking of all places remains positive. Many examples using this semantics are given in [49] while a net system using both semantics is studied in [123].

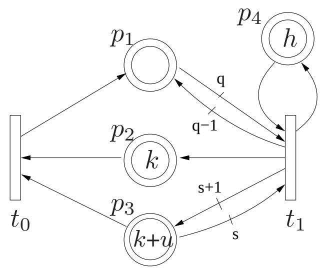
Observe that Equation (4) is not defining completely the flow when there are conflicting transitions. In such a case, a resolution policy should be specified, otherwise many solutions are possible [12]. Moreover, an important drawback of this semantics is that it allows infinitely fast movement of tokens when several transitions in sequence are weakly enabled. In the following, finite server semantics is only considered in Subsection 6.7, the rest of the paper focuses on infinite server semantics.

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

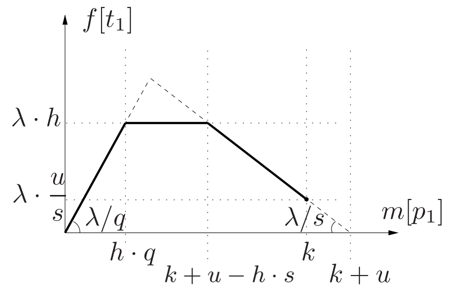
$$f_j = \lambda_j \cdot enab(t_j, \boldsymbol{m}) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \frac{m_i}{\boldsymbol{Pre}[p_i, t_j]}, \quad (5)$$

where λ_j is the firing rate of t_j.

Like in Markovian PNs, i.e., discrete PNs with exponential firing times in all transitions [130], in continuous PNs under *infinite server semantics*, the flow through a transition is proportional to its enabling degree. The dynamical system corresponds to a *piecewise linear* system and switches occur due to the minimum operators. Nevertheless, more complex behaviors can be modeled. For example, in certain application domains as road traffic systems different shapes may be convenient. In this case, it is important to approximate the fundamental diagram expressing the relationship between the density of cars and the flow of cars in a given section of the road. Using some additional places, the flow of a transition can be modeled as a *piecewise linear* function of the marking of a given place [97]. For example, a three phases flow can be easily represented: with an ascending phase, a constant phase (using self-loop places around transitions) and a descending phase (using complementary places). In *Timed differentiable Petri nets* (TDPNs) the idea is analogous to have two separate channels (like in Forrester Diagrams): one devoted to define how tokens flow, i.e., the material flow, and the other to fix the value of the flow, i.e., the information flow. In [138] it is proved that TDPNs can be simulated by TCPNs, having equal modeling capabilities. From a different perspective, an extension of the infinite server semantics is defined in [83] where lower and upper bounds are given for the firing rates. The idea is that using *interval firing speeds* the variability of the stochastic behavior of the underlying discrete model can be taken into account in performance evaluation tasks.



(a) A continuous Petri net with several self-loops.



(b) The flow of transition t_1 (see fig. 6(a)) is a piecewise linear function of the marking of p_1.

Figure 6: Modeling of the flow of a transition as a piecewise linear function of the marking of a given place.

TCPNs under infinite server semantics have the capability to simulate Turing machines [138], thus they have an important expressive power; nevertheless, certain important properties are *undecidable* (for example, marking coverability, submarking reachability or the existence of a steady-state). Moreover, through discoloration of colored nets, the *minimum* operator of infinite server semantics becomes a *product* (population semantics) [153]. Being possible to define firing flows proportional to the product of the marking of input, places *chaotic* models can be described, i.e., models of deterministic dynamical systems that are extremely sensitive to initial conditions.

A model in which the time is associated to places has been introduced in [41]. Under some assumptions on the net structure and on the firing policy, it is equivalent to a linear system in the $(min, +)$ semiring. Unfortunately, basically marked graphs can be studied in this algebra. Following this work, in [68] some results regarding the steady-state have been proved for a particular class of deterministically timed nets under a *stationary routing* (STAR), by which the behavior is constrained to be conflict-free.

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, *finite server* (or *constant speed*) or *infinite server* (or *variable speed*) are mainly used [49, 155]. 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 [49], the authors observed that most frequently the infinite server semantics approximates better the marking of the discrete net system. Moreover, for mono-T-semiflow reducible net systems [100] under some general conditions it is proved that infinite server semantics approximates better the flow in steady state [123]. The result holds depending on an structural property defined from the steady-state marking, a condition that is quite common in the case of production systems. For population systems (predator/prey, biochemistry, . . .), the transition firing flows are usually described by *products* of markings, and even more specific non-linear functions (see, for example, [152] and [78]). In fact, the products can be obtained from infinite server semantics while considering discoloration of colored PN models [152].

4.2 Logical properties in timed models versus untimed models

If the steady-state exists, from (3) and (5), $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f}_{ss} = 0$ is obtained (independently of the firing semantics), where $\boldsymbol{f}_{ss}$ is the flow vector of the timed system in the steady state, $\boldsymbol{f}_{ss} = \lim_{\tau\to\infty} \boldsymbol{f}(\tau)$. Therefore, the flow in the steady state is a T-semiflow of the net. Deadlock-freeness and liveness definitions of untimed systems can be easily extended to timed systems as follows:

Definition 10 *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a timed continuous PN system and $\boldsymbol{f}_{ss}$ be the vector of flows of the transitions in the steady state.*

- $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ *is* timed-deadlock-free *if $\boldsymbol{f}_{ss} \neq \boldsymbol{0}$;*
- $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ *is* timed-live *if $\boldsymbol{f}_{ss} > \boldsymbol{0}$;*
- $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ *is* structurally timed-live *if $\exists\, \boldsymbol{m}_0$ such that $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ is* timed-live.

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 systems to untimed systems. The reverse is not true, i.e., the untimed system can deadlock, but a given $\boldsymbol{\lambda}$ can *drive* the marking along a trajectory without deadlocks, e.g., the system in fig. 4(b) deadlocks as untimed but is timed-live with $\boldsymbol{\lambda} = [1\ 2]^T$ (in particular $\boldsymbol{f}_{ss} = [1\ 1]^T$). 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, which are not necessarily satisfied by the untimed system [168]. The relationships among liveness definitions are depicted in fig. 7.

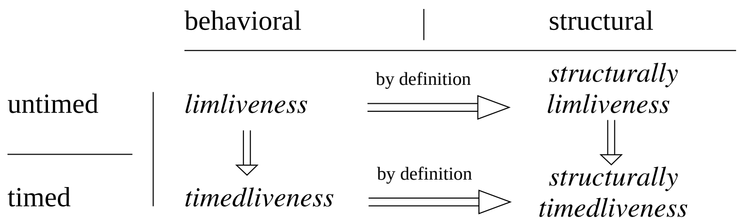


Figure 7: Relationships among liveness definitions for continuous models

As an example, let us show how some conditions initially obtained for timed systems can be applied on untimed ones. It is known that if a MTS timed system $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is structurally live for any $\boldsymbol{\lambda} > 0$, then for every transition t there exists $p \in {}^\bullet t$ such that $p^\bullet = \{t\}$, i.e., p is *persistent* or *conflict-free* [101]. Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a timed system containing a transition t such that for every $p \in {}^\bullet t$, $|p^\bullet| > 1$. According to the mentioned condition $\boldsymbol{\lambda}$ exists such that $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is not structurally timed-live. Therefore $\mathcal{N}$ is not structurally $\lim$-live, since structurally timed-liveness is a necessary condition for structurally $\lim$-liveness (see fig. 7).

4.3 Infinite Server Semantics: Performance bounds for steady-state

The existence of the minimum operator in infinite server semantics induces three strongly related concepts: a) the set of places defining the enabling degree of transitions is known as *configuration*; b) the state space in which the configuration is the same is known as *region*; c) at each region the dynamics are driven by a single *linear system* which is also known as *operation mode*. More formally:

Definition 11 *A* configuration *of a net $\mathcal{N}$ is a set of (p, t) arcs, one per transition,* covering *the set T of transitions. Associated to a given configuration $\mathcal{C}_k$ is the following $|T| \times |P|$ configuration matrix:*

$$\boldsymbol{\Pi}_k[t, p] = \begin{cases} \frac{1}{\boldsymbol{Pre}[p,t]}, & \textit{if } (p, t) \in \mathcal{C}_k \\ 0, & \textit{otherwise} \end{cases} \tag{6}$$

In the case of a TCPN system under infinite server semantics, at a given marking $\boldsymbol{m} \in RS(\mathcal{N}, \boldsymbol{m}_0)$, the flow of a transition t_j, given by (5), is defined by the marking of an input place $p_i \in {}^\bullet t_j$, the one which gives the minimum. Let us notice that the reachability set $RS(\mathcal{N}, \boldsymbol{m}_0)$ of a TCPN system can be partitioned (except on the borders) according to the configurations and inside each obtained *convex region* $\mathcal{R}_i(\mathcal{N}, \boldsymbol{m}_0)$ the system dynamic is linear. Putting together (6), (5) and (3), the dynamic system evolution inside a region $\mathcal{R}_k$, called *operation mode* k as well, can be written as:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{f}(\tau) = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m}(\tau), \tag{7}$$

where $\boldsymbol{\Lambda} = diag(\boldsymbol{\lambda})$ is a diagonal $|T| \times |T|$ matrix containing the firing rates of transitions and the configuration matrix is $\boldsymbol{\Pi}(\boldsymbol{m}) = \boldsymbol{\Pi}_k$ where $\boldsymbol{\Pi}_k$ is the configuration matrix associated to $\mathcal{R}_k$ (if $\boldsymbol{m}$ is on the border of two regions $\mathcal{R}_1$ and $\mathcal{R}_2$, any operation mode with $\boldsymbol{C\Lambda\Pi}_1$ or $\boldsymbol{C\Lambda\Pi}_2$ can be used since the same behavior is obtained).

Obviously, inside a region (where $\boldsymbol{\Pi}(\boldsymbol{m})$ is the current matrix) (7) is a linear system and will be said that it is the k^{th} *linear system* or the k^{th} *(operation) mode* of the TCPN system. To each configuration, an *operation mode* can be associated. The number of modes (regions, configurations) is upper bounded by $\prod_{t \in |T|} |^\bullet t|$ but some of them can be redundant and can be removed [124].

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 steady state. A classical concept in queueing network theory is the "visit ratio". In Petri net terms, the visit ratio of transition t_j with respect to t_i, $\boldsymbol{v}^{(i)}[t_j]$, is the average number of times that 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 $\boldsymbol{m}_0$ (Prop. 3, $\exists \boldsymbol{x} \geq \boldsymbol{1}$, $\boldsymbol{C} \cdot \boldsymbol{x} = 0$ and $\nexists \boldsymbol{y} \geq 0, \boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \leq \boldsymbol{0}, \boldsymbol{y}^T \cdot \boldsymbol{m}_0 = 0$). In order to simplify the presentation, let us assume that the net is MTS. Therefore, for any t_i, $\boldsymbol{f}_{ss} = \chi_i \cdot \boldsymbol{v}^{(i)}$, with χ_i the throughput of t_i. The vector of visit ratios is a right annuler of the incidence matrix $\boldsymbol{C}$, and therefore, in MTS systems, proportional to the unique T-semiflow. For this class of systems, the throughput can be computed using the following non-linear programming problem that maximize the flow of a transition (in fact, any of them, since all are related by the T-semiflow)

$$\begin{array}{ll} \max & \boldsymbol{f}_{ss}[t_1] \\ \text{s.t.} & \boldsymbol{m}_{ss} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{f}_{ss}[t_j] = \lambda_j \cdot \min\limits_{p_i \in {}^\bullet t_j} \left\{ \frac{\boldsymbol{m}_{ss}[p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}, \forall t_i \in T \\ & \boldsymbol{C} \cdot \boldsymbol{f}_{ss} = 0 \\ & \boldsymbol{m}_{ss}, \boldsymbol{\sigma} \geq 0 \end{array} \tag{8}$$

where $\boldsymbol{m}_{ss}$ is the steady-state marking. A way to solve (8), which due to the minimum operator is non linear, consists in using a branch & bound algorithm [100]. 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 transitions for which the flow equation is not satisfied. If the net is not MTS, similar developments can be done by adapting the equations in [38].

In the case of controlled systems, the LPP relaxation of (8) can be used to compute an optimal steady-state assuming only flow reduction (the machines can only be slowed down), $\boldsymbol{f} \geq 0$ and the steady-state flow should be repetitive, $\boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{0}$. If all transitions are controllable, it can be solved by 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 [122]. For example, let us consider the following LPP:

$$
\begin{array}{lll}
\max & \boldsymbol{k}_1 \cdot \boldsymbol{f} - \boldsymbol{k}_2 \cdot \boldsymbol{m} - \boldsymbol{k}_3 \cdot \boldsymbol{m}_0 & \\
\text{s.t.} & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} & \text{(a)} \\
 & f_i = \lambda_i \cdot \left(\frac{\boldsymbol{m}[p_j]}{\boldsymbol{Pre}[p_j, t_i]} \right) - \boldsymbol{v}[p_j, t_i], \forall p_j \in {}^{\bullet}t_i, \boldsymbol{v}[p_j, t_i] \geq 0 & \text{(b)} \\
 & \boldsymbol{C} \cdot \boldsymbol{f} = 0 & \text{(c)} \\
 & \boldsymbol{m}, \boldsymbol{\sigma}, \boldsymbol{f} \geq 0 &
\end{array}
\qquad (9)
$$

where $\boldsymbol{v}[p_j, t_i]$ are *slack* variables. The objective function represents the profit that has to be maximized where $\boldsymbol{k}_1$ is a price vector w.r.t. steady-state flow $\boldsymbol{f}$, $\boldsymbol{k}_2$ is the work in process (WIP) cost vector w.r.t. the average marking $\boldsymbol{m}$ and $\boldsymbol{k}_3$ represents depreciations or amortization of the initial investments over $\boldsymbol{m}_0$. Using the slack variables $\boldsymbol{v}$, the optimal control in steady-state for a transition t_i if it is controllable, i.e., it permits a control $u_i > 0$, is just $u_i = \min_{p_j \in {}^{\bullet}t_i} \boldsymbol{v}[p_j, t_i]$. 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 a dynamic perspective in section 6.

4.4 Infinite server semantics: monotonicity and paradoxes

According to (5), 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 (5) as well, two homotheticity properties are dynamically satisfied:

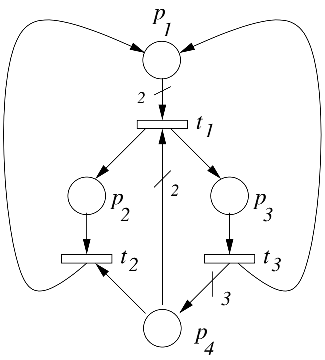
- if $\boldsymbol{\lambda}$ is multiplied by a constant $k > 0$ then identical markings will be reached, but the system will evolve k times faster;
- if the initial marking is multiplied by k, the reachable markings are multiplied by k and the flow will also be k times bigger.

Unfortunately, infinite server semantics has not only "good" properties and some counterintuitive behaviors or "paradoxes" appear. For example, it could be thought that, since fluidization relax some restrictions, the throughput of the continuous system should be at least that of the discrete one. However, the throughput of a TCPN is not in general an upper bound of the throughput of the discrete PN; moreover, if only some components of $\boldsymbol{\lambda}$ or only some components of $\boldsymbol{m}_0$ are increased the steady state throughput is not monotone in general [155].

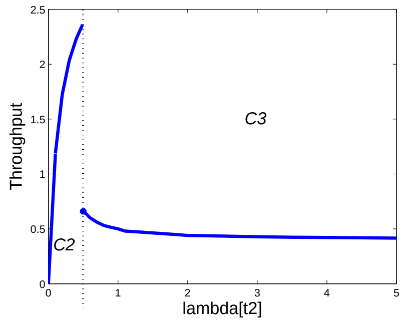
Two monotonicity results of the steady-state throughput are satisfied under some general conditions [123]:

Proposition 12 *Assume* $\langle \mathcal{N}, \lambda_i, \boldsymbol{m}_i \rangle$, $i = 1, 2$ *are MTS TCPNs under infinite server semantics that 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 for* $\boldsymbol{m} \in [\underline{\boldsymbol{m}}, \overline{\boldsymbol{m}}]$ *and* $\boldsymbol{\lambda} \in [\underline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\lambda}}]$*. Then for all* $\boldsymbol{m}_1, \boldsymbol{m}_2 \in [\underline{\boldsymbol{m}}, \overline{\boldsymbol{m}}]$ *with* $\boldsymbol{m}_1 \leq \boldsymbol{m}_2$ *and for all* $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in [\underline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\lambda}}]$ *with* $\boldsymbol{\lambda}_1 \leq \boldsymbol{\lambda}_2$*, the steady state flows satisfy* $\boldsymbol{f}_1 \leq \boldsymbol{f}_2$.

Let us consider the mono-T-semiflow TCPN in fig. 8(a) under infinite server semantics with $\lambda_1 = \lambda_3 = 1$ and $\boldsymbol{m}_0 = [15\ 1\ 1\ 0]^T$. Different modes can govern the evolution of the system at steady-state. For example, if $0 < \lambda_2 \leq 0.5$, the flow in



(a) A Mono-T-semiflow net.



(b) Throughput for $\boldsymbol{m}_0 = [15\ 1\ 1\ 0]^T$ and different values of λ_2

Figure 8: 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).

steady-state is $f_1(\tau) = m_1(\tau)$, $f_2(\tau) = m_4(\tau)$ and $f_3(\tau) = m_3(\tau)$, respectively. Therefore, $\mathcal{C}_2 = \{(p_1, t_1), (p_4, t_2), (p_3, t_3)\}$ is the steady-state configuration and the set of places $\{p_1, p_4, p_3\}$ gives the flow. Since it contains the support of a P-semiflow ($p_1 + 4 \cdot p_3 + p_4$) the steady-state flow is monotone (fig. 8(b)). Increasing λ_2, the steady-state configuration becomes $\mathcal{C}_3 = \{(p_4, t_1), (p_2, t_2), (p_3, t_3)\}$, i.e., the set of places governing the evolution becomes $\{p_4, p_2, p_3\}$, that is the support of a P-flow ($p_2 - 3 \cdot p_3 - p_4$), not a P-semiflow, and monotonicity may not hold (fig. 8(b)).

4.5 Infinite server semantics: analyzing by model checking

Born in the Computer Science milieu, *model checking* techniques are very popular for formal verification of DEDS (see for example [40], the ACM Turing Award of 2007). Given a model and a specification, *model checking* tests automatically whether the model meets the specification or not. Herein we deal with continuous systems, therefore we need a "discrete view". For a TCPN system under infinite server semantics formal analysis starts by embedding the TCPN system into a piecewise affine (PWA) system and then into a finite transition system based on discrete abstractions (finite quotients). The obtained quotient is iteratively analyzed and refined by employing convexity properties of affine systems in full-dimensional polytopes [107].

Let us assume that $\mathcal{P}$ is a user-defined set of strict linear inequalities over marking $\boldsymbol{m}$, including all the affine functions in $\boldsymbol{m}$ necessary to define the full-dimensional regions $\mathcal{R}_i$. Two kinds of interesting problems can be formulated as:

- *Construction of safe sets*: let us assume a given set of initial markings defined as the conjunction of predicates from a set $\mathcal{P}_0 \subseteq \mathcal{P}$. The problem is to find a subset of the reachability set that cannot be reached by trajectories of TCPN originating in the initial set.

- *Initial set satisfying Linear temporal logic (LTL) specification*: given an LTL formula over $\mathcal{P}$, find a set of initial markings of TCPN from where all possible trajectories satisfy the formula.

The basic idea for the formal verification of PWA systems is based on the results in [73]. Given two adjacent polytopes, it is shown that there exists a trajectory penetrating from one to another in finite time if and only if there exists a vertex on the common facet at which the projection of the vector field on the outer normal of the facet pointing from the first to the latter is strictly positive. Moreover, it is proved also that an affine system has a trajectory contained in a full dimensional open polytope for all time if and only if the affine system has an equilibrium inside the polytope. Therefore, solving both problems of TCPN reduces to checking nonemptiness of polyhedral sets.

A fully automated procedure is proposed in [108] for both problems. Since the procedure basically reduces to search on a graph, the complexity is dependent on the number of nodes, each node corresponding to a region in the obtained partition. Thus, the bottleneck is resulting from the refinement procedure because after each refinement the number of discrete states in the transition system is increasing.

4.6 On the approximation by fluidization

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 [49, 139]. Nevertheless, the dynamic behavior of a timed continuous PN model does not always approximate that of the corresponding timed discrete PN. Then, it is important to investigate the conditions that lead to a valid relaxation, from the performance evaluation perspective. In some sense, this subsection deals with the *legitimization* of the so called infinite server semantics (introduced in subsect. 4.1) and the consideration of some issues that affect the quality of the approximation. The following subsection considers a few techniques for *improving* the approximation.

Let us consider Markovian Petri nets (MPN), i.e., stochastic discrete Petri net with exponential delays associated to the transitions and conflicts solved by a race policy [130]. The approximation of the average marking of an ergodic (thus with home states) MPN, by that of the corresponding TCPN under *infinite server semantics* (ISS), was first considered in [140], later more deeply studied in [165]. In this last work it is concluded that the approximation holds when the utilization factor is high, usually when the number of active servers of transitions (the probability of being enabled) is large as well, and the system mainly evolves inside one marking region, 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* (in non-ordinary nets) and *joins* (*rendez-vous*). The reason is that the flow definition for the TCPN does not accurately describe the throughput in these cases. In fact, the approximation is perfect for ordinary Join-Free Petri nets.

Let us provide an intuitive reasoning for this. As previously stated, $\boldsymbol{M}$ and $Enab$ refer to the discrete MPN, while $\boldsymbol{m}$ and $enab$ refer to the continuous model, TCPN. Suppose that, at some time, the marking of the TCPN approximates the average marking of the MPN, i.e., $\boldsymbol{m} \sim E\{\boldsymbol{M}\}$. Given an arc with weight k connecting a place p_j to a transition t_i, the *expected* enabling degree of t_i in the MPN would be $E\{Enab(t_i)\} = E\{\lfloor \boldsymbol{M}[p_j]/k \rfloor\}$, which is different than the enabling degree in the TCPN $enab(t_i) = \boldsymbol{m}[p_j]/k \sim E\{\boldsymbol{M}[p_j]\}/k$, due to the presence of the operator $\lfloor \cdot \rfloor$ (in ordinary arcs $k = 1$, thus $\lfloor \boldsymbol{M}[p_j]/k \rfloor = \boldsymbol{M}[p_j]$). Similarly, given a synchronization t_i with two input places $\{p_j, p_k\}$, the expected enabling in the MPN would be $E\{Enab(t_i)\} = E\{min(\boldsymbol{M}[p_j], \boldsymbol{M}[p_k])\}$, which is not equal to the enabling in the TCPN $enab(\boldsymbol{m}[p_j], \boldsymbol{m}[p_k]) \sim min(E\{\boldsymbol{M}[p_j]\}, E\{\boldsymbol{M}[p_k]\})$, because the "expected value" and the "min" operators

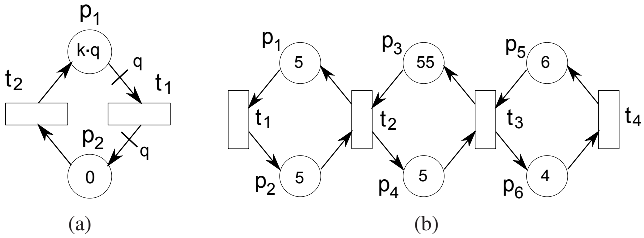


Figure 9: a) Cycle net with arc weights. b) Marked graph that evolves through different regions.

Table 2: Throughput and its approximation for t_1 of the net of fig. 9(a).

$k = 1 \setminus q =$	1	2	4	8	16
MPN, $\boldsymbol{\chi}[t_1]$	0.50	0.40	0.32	0.26	0.22
TCPN, $\boldsymbol{f}[t_1]$	0.50	0.50	0.50	0.50	0.50
$q = 4 \setminus k =$	1	2	4	8	16
MPN, $\boldsymbol{\chi}[t_1]$	0.32	0.80	1.76	3.78	7.68
TCPN, $\boldsymbol{f}[t_1]$	0.50	1.00	2.00	4.00	8.00

do not commute (also seen in the context of performance evaluation of throughput bounds [27]). Consequently, since the flow through the transitions depends on the enabling degree, a perfect approximation will not hold for future time. Nevertheless, approximation errors do not accumulate when 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.

Let us illustrate with an example how the arc weights introduce approximation errors. Consider the MPN system of fig. 9(a) with timing rates $\lambda_1 = \lambda_2 = 1$, and initial marking $\boldsymbol{M}_0 = [k \cdot q\ 0]^T$, 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 2. Note that, when $k = 1$, i.e., the marking is relatively very small, the larger the weight of the input arc of t_1, i.e., q, the bigger the error between the throughput in the MPN ($\boldsymbol{\chi}[t_1]$), and the flow in the TCPN ($\boldsymbol{f}[t_1]$). Observe that the flow in the continuous model remains unchanged. Actually, it is very important to remark that the differential equation describing the behavior of the TCPN does *not* depend on q:

$$\overset{\bullet}{\boldsymbol{m}} = \boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\boldsymbol{m})\boldsymbol{m} = \begin{bmatrix} -q & 1 \\ q & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{q} & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{m} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \boldsymbol{m} \tag{10}$$

On the other hand, when the arc weights are fixed but the initial marking is increased, i.e., k is increased, the relative approximation error decreases (in such case, $E\{\lfloor \boldsymbol{M}[p_1]/q \rfloor\} \sim E\{\boldsymbol{M}[p_1]\}$ for $\boldsymbol{M}[p_1] >> q$). Concluding: the relative errors introduced by arc weights become smaller when the marking in the net is increased w.r.t.

Table 3: Marking approximation of p_3 for the MPN of fig. 9(b) (which is a marked graph, so, state machine decomposable).

λ_4	MPN	$TCPN$	$TnCPN$	$E\{Enab(t_4)\}$	$Prob(\boldsymbol{M} \in \mathcal{R}_{ss})$
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

those weights.

Now, let us illustrate how the joins also introduce approximation errors. The MPN system of fig. 9(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 models were also simulated. The average markings at the steady state are shown in table 3 (columns MPN and TCPN). The column TnCPN corresponds to the stochastic extension of the TCPN obtained by adding some gaussian *noise*, something to be considered in section 4.8. The column denoted as $E\{Enab(t_4)\}$ is the average enabling degree of t_4 in the MPN at the steady state ($\forall t_i \in T \setminus \{t_4\} : E\{Enab(t_4)\} < E\{Enab(t_i)\}$ in all the experiments, so this represents a lower bound for the number of active servers in the transitions). The value in column $Prob(\boldsymbol{M} \in \mathcal{R}_{ss})$ is the probability that the marking is inside the region $\mathcal{R}_{ss}$, related to the steady state of the TCPN (equivalently, the fraction of time that $\boldsymbol{M}(\tau)$ is in $\mathcal{R}_{ss}$). Note that the lower the probability that $\boldsymbol{M}(\tau)$ belongs to $\mathcal{R}_{ss}$, 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 $\boldsymbol{M}(\tau) \in \mathcal{R}_{ss}$ is high, which occurs for $\lambda_4 = 2$. The approximation holds because, in this case, $\boldsymbol{M}(\tau)$ mainly evolves in one region $\mathcal{R}_{ss}$ (in particular, $E\{min(\boldsymbol{M}[p_4], \boldsymbol{M}[p_5])\} \sim E\{\boldsymbol{M}[p_4]\}$ and $E\{min(\boldsymbol{M}[p_2], \boldsymbol{M}[p_3])\} \sim E\{\boldsymbol{M}[p_2]\}$), where the continuous model has an asymptotically stable steady sate marking.

An analogous approximation analysis has been recently achieved in the framework of fluid process algebra ($PEPA$), deriving a functional limit theorem [55]. That approach is based on classic works where ordinary differential equations are used for describing the transient behavior of the limit of a sequence of Markov processes [110]. The resulting theorem establishes that, with a randomly high probability, the relative distance between the state of the discrete and of the fluid systems becomes arbitrarily small, during a *finite time* interval, when the number of initial process-algebra components is increased towards infinity. In Petri nets this would be expressed as $\forall \delta > 0, T < \infty$: $lim_{k \to \infty} Prob\{\frac{1}{k} ||\boldsymbol{M}\ (\tau, \boldsymbol{M}_0 \cdot k) - \boldsymbol{m}(\tau, \boldsymbol{m}_0 \cdot k)|| < \delta | \, \tau \in [0, T]\} = 1$. It is very important to remark that the analysis is restricted to a *finite* time interval. Nevertheless, the approximation in finite time does not imply the approximation in *steady state* (a more interesting issue from our perspective), specially when the initial number of components is increased leading to a larger transient behavior. In the $PEPA$ framework, the steady state approximation has been recently studied [77]. Assuming a unique stationary state in the underlying Markov chain of the discrete model and the existence of a globally asymptotically stable equilibrium point (called fixed point) in the fluid one, it is showed that the relative distance between the steady state of both systems becomes zero when the number of initial components is increased towards infinity. The analysis becomes more complex if the stability precondition is not fulfilled. Even more, the approximation may not hold.

Consider for instance the (discrete) PN system of fig. 9(b) with $\lambda_4 = 1$ and initial marking $\boldsymbol{M}_0 = k \cdot [5\ 5\ 55\ 5\ 5\ 5]^T$. For any $k \in \mathbb{N}^+$, the average steady state is $E\{\boldsymbol{M}_{ss}\} = k \cdot [5\ 5\ 30\ 30\ 5\ 5]^T$ (known by the system's symmetry). On the other hand, the steady state for the TCPN is $\boldsymbol{m}_{ss} = \boldsymbol{m}_0 = k \cdot [5\ 5\ 55\ 5\ 5\ 5]^T$, since this is an equilibrium point in the TCPN (but not the only one). Then, it is clear that the TCPN system does not approximate the MPN for any value of k. In fact, the average relative distance can be directly computed as $\frac{1}{k}||\boldsymbol{M}(\tau, \boldsymbol{M}_0 \cdot k) - \boldsymbol{m}(\tau, \boldsymbol{m}_0 \cdot k)|| = 35.35$, $\forall k \in \mathbb{N}^+$. On the other hand, considering a finite time interval $T < \infty$, it is always possible to set a large enough value for k s.t. the average relative distance is arbitrarily small during $\tau \in [0, T]$. Intuitively: increasing the initial marking makes the transient behavior to be longer, so, when $k \to \infty$ while T remains constant has a similar effect that keeping k constant while $T \to 0$ (thus $\boldsymbol{M}(T) \to \boldsymbol{M}_0$, and $\boldsymbol{m}(T) \to \boldsymbol{m}_0$).

A couple of different examples can be found in [153]. The first one is the "gambler's ruin problem", represented as a stochastic Petri net. In this, a deadlock is reached with probability one. Nevertheless, by multiplying the initial state by a factor k, and given some particular rates (providing equal probabilities to the transitions in conflict), the transient behavior, i.e., the average time to reach the deadlock, becomes larger with a higher rate, of order of k^2. Consequently, if a finite time interval is considered, the two deadlocks of the (discrete) system may be ignored when k is too large. The second example is the well known predator/prey model of Volterra-Lotka, which can be modelled as a colored PN system. This model is fluidified leading to the product server semantics. The discrete PN model is unbounded and non-live [134], while the continuous is bounded and live! (the system describes an orbit in the phase portrait [153]). Since the transient behavior of the discrete model may be very large, this can be approximated by the continuous model for a very long time. Nevertheless, sooner or later, the discrete model will become non-live (all the predators will die), while the continuous system will remain live. Even more, if all the predators die before the prey, these can grow unboundedly, while the continuous approximation will remain bounded.

4.7 Improving the approximation: removing spurious solutions, addition of noise, modification of the semantics

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 subsection, three different approaches, for such improvement, will be discussed.

Removing spurious solutions. In subsection 3.3 it was pointed out that spurious solutions become reachable markings in the autonomous continuous model, affecting the quality of the fluidization. This is specially undesirable when the spurious solutions represent deadlocks in the continuous PN while the discrete system is live. This problem may also appear in the timed continuous model. Even in the case that the spurious deadlock is not reachable by the TCPN system, i.e., it is reachable for the autonomous continuous, but not for the timed continuous given the particular initial marking and timing, the existence of such deadlock marking in the autonomous net affects the dynamic behavior of the TCPN. 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.

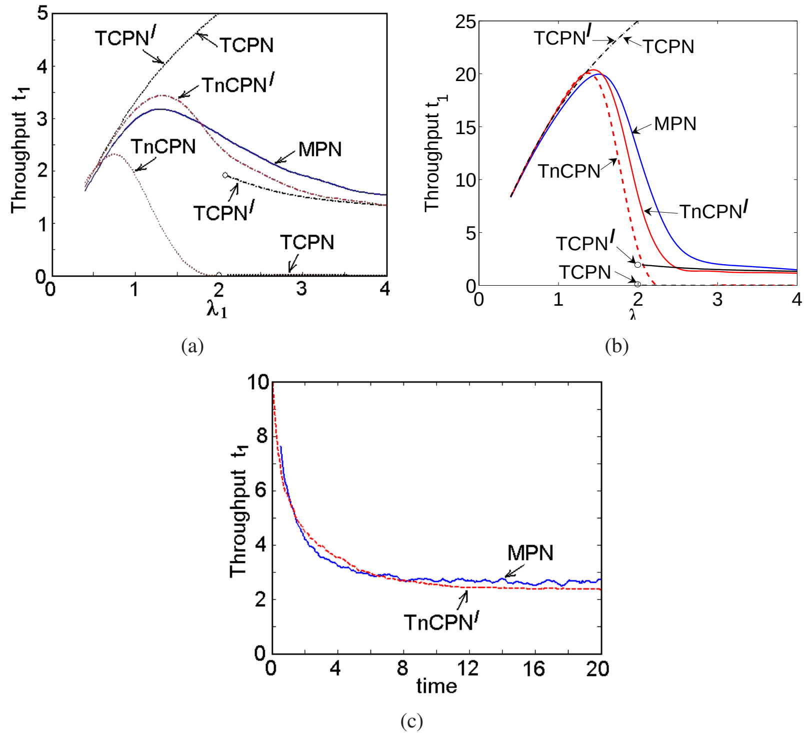


Figure 10: Throughput for the MPN system of fig. 5 and its corresponding continuous relaxations, for rates $\lambda_1 \in [0.4, 4]$, $\lambda_2 = 1$ and initial marking a) $\boldsymbol{M}_0 = [10\,11\,0]^T$ and b) $\boldsymbol{M}_0 = [50\,55\,0]^T$. In all figures, $TCPN'$ and $TnCPN'$ represent the fluid models in which the spurious deadlock has been previously removed. There is a discontinuity in TCPN and $TCPN'$ at $\lambda_1 = 2$. c) Average transient behavior of the throughput at t_1 for $\boldsymbol{M}_0 = [10\,11\,0]^T$ and $\lambda_1 = 2$.

As an example, consider the MPN given by the net of fig. 5 with initial marking $\boldsymbol{M}_0 = [10\,11\,0]^T$ and rates $\boldsymbol{\lambda} = [0.4\,1]$. As shown in subsection 3.3, this PN has a spurious deadlock, which can be removed by eliminating the two (discrete) *frozen tokens* from p_2. This is equivalent to consider $\boldsymbol{M}'_0 = [10\,9\,0]^T$ as the initial marking. The MPN and the corresponding fluid model TCPN have been simulated for both initial markings $\boldsymbol{M}_0$ (with spurious deadlocks) and $\boldsymbol{M}'_0$, 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. 10(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 $\boldsymbol{m}_0 = \boldsymbol{M}_0$ 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 (with $\boldsymbol{M}'_0$), for any $\lambda_1 > 2$ (for $\lambda_1 \leq 2$ there is no change in the TCPN).

Stochastic continuous PN. The approximation of the average marking of an ergodic Markovian Petri net may be improved by adding white noise to the transitions flow of the TCPN [165]. Intuitively speaking, the transitions firings of a MPN are stochastic processes, then, the noise added to the flow in the TCPN may help to better approximate such stochastic behavior, which is particularly relevant at the synchronizations. The model thus obtained (here denoted as TnCPN) is represented, in discrete time, as: $\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C}(\boldsymbol{\Lambda\Pi}(\boldsymbol{m}_k)\boldsymbol{m}_k\Delta\tau + \boldsymbol{v}_k)$, with $\boldsymbol{v}_k$ being a vector of independent normally distributed random variables with zero mean and covariance matrix $\sum_{\boldsymbol{v}_k} = diag(\boldsymbol{\Lambda\Pi}(\boldsymbol{m}_k)\boldsymbol{m}_k\Delta\tau)$.[4]

This modification is particularly relevant when the system evolves through 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 $enab(t_i) = min(\boldsymbol{m}[p_i^1], .., \boldsymbol{m}[p_i^k]) \sim min(E\{\boldsymbol{M}[p_i^1]\}, .., E\{\boldsymbol{M}[p_i^k]\})$ may become relatively important). The approximation is improved when the number of active servers is increased, as already said, assuming *asymptotic stability* and *liveness* in the continuous system (thus it is important to remove the spurious deadlocks).

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 of 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. 9(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 3. As it was pointed out in the previous subsection, the lower the probability that $\boldsymbol{M}_k$ belongs to $\mathcal{R}_{ss}$, 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. 5 with $\boldsymbol{M}_0 = [10\,11\,0]^T$. The steady state throughput of the MPN and its different relaxations is shown in fig. 10(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 $\boldsymbol{M}_0 = [10\,9\,0]^T$ (see subsection 3.3), the TnCPN approximates better the MPN than the TCPN model. Fig. 10(b) shows the results of the same experiment but with a bigger population. In this case, $\boldsymbol{M}_0 = 5 \cdot [10\,11\,0]^T = [50\,55\,0]^T$ and the spurious solution is removed by considering the initial marking $\boldsymbol{M}'_0 = [50\,49\,0]^T$ (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., $\boldsymbol{M}'_0 \neq 5 \cdot [10\,9\,0]^T$, then the curve TCPN′ in fig. 10(b) is not in homothetic relation with that in fig. 10(a) (but the original TCPN is). It can be observed in fig. 10(b) that now the continuous models provide a better approximation than in the case of fig. 10(a), because the population is bigger. Finally, fig. 10(c) shows the transient trajectory described by the average throughput of t_1, for the case $\boldsymbol{M}_0 = [10\,11\,0]^T$ 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.

Modification of the semantics. As already mentioned, the existence of arc weights (a kind of lot-synchronization) affects the quality of the fluidification. A critical case occurs when there exists a transition t_j with a q-bounded input place $p_i \in {}^\bullet t_j$ and

[4]For simulation purposes, in the state equation for $\boldsymbol{m}_{k+1}$, the noise $\boldsymbol{v}_k$ is added only if $\boldsymbol{m}_{k+1} \geq 0$. This means that very close to boundaries the system may be kept as deterministic. In fact, if the system is crowded, i.e., $\boldsymbol{m}_0$ is big, the probability of getting $\boldsymbol{m}_{k+1} \not\geq 0$ is very low.

the weight of the arc connecting them is q as well (in any case, since liveness is assumed, the weight of the arcs cannot be larger than the bound of the corresponding input places). Consequently, the marking at p_i must be equal to its upper bound in order to enable t_j (the most basic example of this case is given in fig. 9(a) for $k = 1$, where the TCPN fails in approximating the throughput of t_1 when $q >> 1$). In this situation, transition t_j is enabled only at a few specific markings (in the example of fig. 9(a), the worst case is found because t_1 is enabled at *only one* marking) of the autonomous reachability set. This enabling property is not captured by the continuous relaxation, where t_j is enabled whenever the places in ${}^\bullet t_j$ are marked, leading to significant approximation errors, i.e., $Enab(t_j) = \lfloor \boldsymbol{M}[p_i]/q \rfloor = 0$ for almost all the markings, while in the TCPN $enab(t_j) = \boldsymbol{m}[p_i]/q > 0$ whenever $\boldsymbol{m}[p_i] > 0$. In order to improve the continuous approximation (for hybrid approximation other reasoning must be considered), the server semantics of the TCPN must be modified for t_j. A heuristic way for doing this, assuming $\{p_i\} = {}^\bullet t_j$, consists in the following expression: $\boldsymbol{f}[t_j] = \boldsymbol{\lambda}_i(\boldsymbol{m}[p_i]/q)^q$ (this is obtained from a probabilistic relaxed view, in which the probability of a token to be in p_i is assumed as $E\{\boldsymbol{M}[p_i]\}/q$). This equation is equivalent to $\boldsymbol{f}[t_j] = \boldsymbol{\lambda}_i \boldsymbol{m}[p_i] \cdot (\boldsymbol{m}[p_i]^{q-1}/q^q)$, which can be seen as the original ISS but multiplied by the marking-dependent function $(\boldsymbol{m}[p_i]^{q-1}/q^q)$. This modification may provide a better approximation. For instance, in the net in fig. 9(a) with $k = 1$ and $q = 4$, the throughput of t_1 obtained with this new semantic is 0.275, which is closer to the throughput of the MPN (0.32) than that obtained with the ISS (0.5). Nevertheless, further investigation is required in order to understand how and when the improvement is achieved.

Another semantics-modification approach has been introduced in [114,115]. There, in order to make the steady state of the continuous PN ($\boldsymbol{m}_{ss}$) to coincide with that of the MPN ($\boldsymbol{M}_{ss}$), the authors propose a modification of the firing rates $\boldsymbol{\lambda}$ of the transitions in the continuous model. Two techniques are proposed: in the first one the firing rates are defined as *piecewise-constant*, i.e., $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_r\}$, depending on the configuration at which $\boldsymbol{m}(\tau)$ belongs, while in the second case (called *adaptive*) the firing rates are adjusted according to the instantaneous approximation error (in particular, $\dot{\boldsymbol{\lambda}} = \eta \cdot diag(\beta \boldsymbol{C}^T(\boldsymbol{M}_{ss} - \boldsymbol{m}(\tau)) + (1-\beta)(\boldsymbol{\chi}_{ss} - \boldsymbol{f}(\tau))$, with $\eta > 0$ and $\beta \in [0, 1]$ being decision parameters). Errors may appear in specific configurations, called *critical*. For mono-T-semiflow nets, such critical regions can be avoided by setting the firing rates with a suitable fix value, providing in this case a homothetic approximation of the average steady state marking and throughput of the MPN, i.e., $\boldsymbol{M}_{ss} = \alpha \boldsymbol{m}_{ss}$ and $\boldsymbol{f}_{ss} = \alpha \boldsymbol{\chi}_{ss}$.

5 Observability and observers

Reconstructing the state of a system from available measurements is a fundamental issue in system theory which may be considered as a self-standing problem, or it can be seen as a pre-requisite for solving a problem of different nature, such as stabilization, state-feedback control, diagnosis, filtering, and others. In the case of CPNs, this problem has been studied for both untimed and timed models under infinite server semantics. In the case of untimed systems (see Section 5.4), the state estimation is conceptually and methodologically closer to the one of discrete event systems since the firing of transitions can be assumed/seen as sequential and the corresponding events not appearing simultaneously. In this case, it is assumed that some events (transitions) are not observable and the initial marking known. The problem is to estimate the possible marking after each observable event (transition). For timed systems, the problem has been studied mainly for infinite server semantics and, since the evolution can be characterized by a set of switching linear differential equations, the state estimation problem is more related to the linear and hybrid system theory. It is a better informed model because of the time constraints, here it is assumed that the initial marking is unknown. Measuring the amount of tokens is some places, the problem is to estimate the current and initial marking of the net.

5.1 On three conceptual levels for timed systems under infinite server semantics: observability, generic observability and structural observability

Three different concepts of observability can be defined for TCPN based on the knowledge of the firing rate vector. Assuming a constant value for the firing vector and measuring a subset of places, the "classical" observability problem is to estimate the initial state/marking. In this case, the set of differential equations is fixed and the concept is called *observability in infinitesimal time*. Observability criteria of *piecewise affine systems* can be applied to TCPN since this is a subclass of those systems. It is well known that the observability in this case is a more difficult problem that the one of linear systems because not only the observability of continuous states is required, but also that of the discrete states [14]. It should be always possible to say which is the linear system governing the evolution.

Let us assume that we can attach some sensors to a set of places $P_o \subseteq P$, the token load of these places being measured at every time instant. Frequently, the marking of some places are impossible to be measured either due to the fact that the sensor is too expensive or because of the physical nature of the state. The problem is to estimate the marking of the other places $P \setminus P_o$. Going back to (7), the system considered here is given by:

$$\left\{ \begin{array}{l} \dot{\boldsymbol{m}}(\tau) = \boldsymbol{A}_i \cdot \boldsymbol{m}(\tau), \boldsymbol{m} \in \mathcal{R}_i \\ \boldsymbol{y}(\tau) = \boldsymbol{S} \cdot \boldsymbol{m}(\tau) \end{array} \right. \tag{11}$$

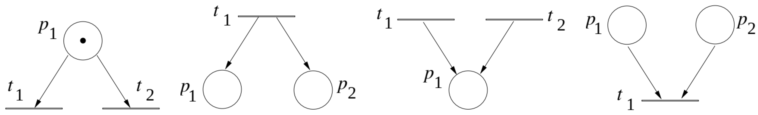
where $\boldsymbol{A}_i = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}_i$ and $\boldsymbol{S}$ is a $|P_o| \times |P|$ matrix, each row of $\boldsymbol{S}$ has all components zero except the one corresponding to the j^{th} measurable place that is 1.

Definition 13 *Let $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system with infinite server semantics and $P_o \subseteq P$ be the set of measurable places. Σ is* observable in infinitesimal time *if it is always possible to compute its initial state $\boldsymbol{m}_0$ in any time interval $[0, \epsilon)$, $\forall \epsilon > 0$.*

In many real systems, the possibility to estimate/observe the system for all possible values of $\boldsymbol{\lambda}$ is an important problem. In this framework, *structural observability* is defined and approaches based on graph-based arguments are used to study it.

Definition 14 *Let $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system with infinite server semantics and $P_o \subseteq P$ be the set of measurable places. Σ is* structurally observable *if it is always possible to compute its initial state $\boldsymbol{m}_0$ in any time interval $[0, \epsilon)$, for all $\epsilon > 0$ and for all $\boldsymbol{\lambda} > \boldsymbol{0}$.*

Finally, if one wants to estimate the system not "for all" but "for almost all" possible values of firing rate, *generic observability* is defined. Also here, graph based approaches are used. This concept is close and inspired from similar works on *linear structured systems* [56].



(a) Choice - p_1 (b) Fork - t_1 (c) Attribution - p_1 (d) Join - t_1

Figure 11: Choices and forks allow backward reasoning to observe the net system. Attributions (in some cases) and joins are problematic.

Definition 15 *Let $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system with infinite server semantics and $P_o \subseteq P$ be the set of measurable places. Σ is* generic observable *if it is always possible to compute its initial state $\boldsymbol{m}_0$ in any time interval $[0, \epsilon)$, for all $\epsilon > 0$ and for all $\boldsymbol{\lambda} > \mathbf{0}$ outside of a proper algebraic variety of the parameter space.*

Obviously, if a TCPN is structurally observable, it is generic observable and observable in infinitesimal time.

5.2 Observability criteria for infinite server semantics

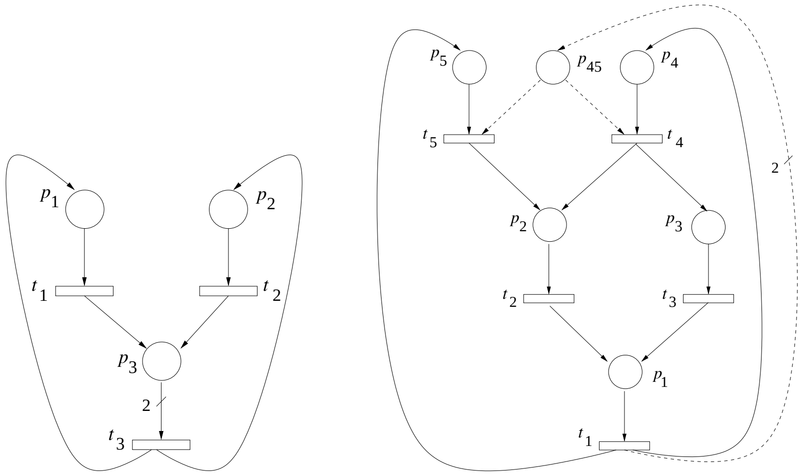
The observability problem can be studied using *graph based approaches* or *matrix algebraic techniques*. Let us see first which are the most important graph based approaches exploring how basic PN constructions (see fig. 11(a)-11(d)) affect the observability of the system. First, let us assume that the net system has only *choices* (fig. 11(a)) and *forks* (fig. 11(b)). If p_i is measured, $m_i(\tau)$ and its variation, i.e., $\dot{m}_i(\tau)$, are known at every time moment τ. Because the net has no joins, the flow of each output transitions t_j of p_i is the product of λ_j and m_i. Knowing the derivative and the output flows, the input flow of the unique (because attributions are not still allowed) input transition t_k can be estimated. Based on the definition of the firing semantics, f_k is the product between λ_k and the marking of ${}^\bullet t_k$. Notice that $|{}^\bullet t_k| = 1$ since there are no joins. Obviously, the marking of ${}^\bullet t_k$ can be computed immediately. Observe that this is a *backward* procedure: measuring p_i, ${}^\bullet({}^\bullet p_i)$ is estimated in absence of joins and attributions.

Therefore, if there exists a path from a place p_i to a measured place p_j not containing any join or attribution then p_i is *structurally observable*, i.e., observable for any values of the firing rates of transitions belonging to the path p_i to p_j [98]. Hence, for net systems without attributions and joins, measuring at least one place from each terminal strongly connected p-component (a subnet generated by a set of places such that there exists a path between each pair of places) the net system is structurally observable [124]. Therefore, it is also *observable* and *generic observable*.

Let us consider now *attributions* and see that this construction can lead to the loss of observability. Assume the TCPN system in fig. 12(a) where p_3 (an attribution place) is the measured place. Writing down the differential equation we have:

$$\dot{m}_3(\tau) = \lambda_1 \cdot m_1(\tau) + \lambda_2 \cdot m_2(\tau) - \lambda_3 \cdot m_3(\tau)$$

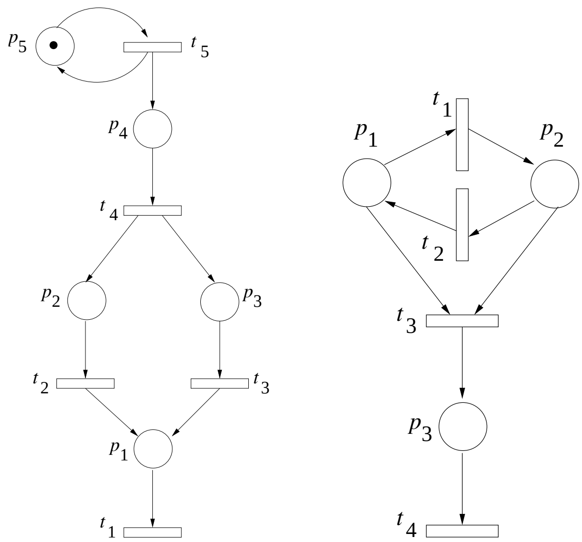
From the previous equation, $\lambda_1 \cdot m_1(\tau) + \lambda_2 \cdot m_2(\tau)$ can be computed since the other variables are known. Nevertheless, if $\lambda_1 = \lambda_2$, it will be impossible to distinguish between $m_1(\tau)$ and $m_2(\tau)$ and the system is not observable. In general, if there exist two transitions with the same firing rate, each one on a different input path in the attribution



(a) A JF net that is not observable if p_3 is measured and $\lambda_1 = \lambda_2$

(b) A JF net that is observable measuring the attribution place p_2 even if $\lambda_4 = \lambda_5 = \lambda_2 = \lambda_3$

Figure 12: Two CPNs with attributions.



(a) A JF net for which if $\lambda_2 = \lambda_3, m_2$ and m_3 cannot be estimated by measuring p_1

(b) A net having both linear systems observable but not overall observable if p_3 is measured and $\lambda_1 = \lambda_2$

Figure 13: Two CPNs with joins.

(paths not containing any attribution or fork), the system is not *observable* [124]. Nevertheless, this is not a general rule since the observability is a global property. Remark also that the system is observable if p_1 is measured even if $\lambda_1 = \lambda_2 = \lambda_3$. In this case, the attribution in p_3 is "destroyed" since the output transitions of the measured places can be removed without affecting the observability space [98].

Let us consider the TCPN is fig. 12(b) with $\boldsymbol{\lambda} = \mathbf{1}$ and assume that p_2 is measured. Then m_4 and m_5 cannot be estimated directly, but their sum (a linear combination of them) is computable (place p_{45} in the figure). Going backwards, m_1 is estimated and, even although m_1 is an attribution, since m_2 is measured, then m_3 can also be estimated. Using m_3, now m_4 is estimated and, through the linear combination of p_{45}, m_5 as well. Therefore, by measuring p_2 the system is structurally observable.

An explanation to the previous loss of observability in join free nets, i.e., linear net models, is obtained by consideration of the transfer functions in Laplace domain. The basic idea is that attributions introduce zeros in the transfer function. Therefore, some pole-zero cancelations may appear, leading to the already mentioned loss of observability. Let us illustrate this with the TCPN system in fig. 13(a), assuming that the attribution p_1 is measured. The system is linear since the net is join-free. If we consider that the input of the system is f_4 and the output is m_1, the transfer function vector between the input flow in places and the output is:

$$\mathcal{Y}(s) = \frac{1}{(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)} H^T \tag{12}$$

where:

$$H = \begin{bmatrix} (s+\lambda_2)\cdot(s+\lambda_3)\cdot(s+\lambda_4) \\ \lambda_2\cdot(s+\lambda_3)\cdot(s+\lambda_4) \\ \lambda_3\cdot(s+\lambda_2)\cdot(s+\lambda_4) \\ (\lambda_2\cdot(s+\lambda_3)+\lambda_3\cdot(s+\lambda_2)) \end{bmatrix} \tag{13}$$

Obviously, if $\lambda_2 = \lambda_3$ or $\lambda_4 = \frac{2\cdot\lambda_2\cdot\lambda_3}{\lambda_2+\lambda_3}$ there is a pole-zero cancellation in *all* elements of vector $\mathcal{Y}(s)$. Therefore, when the net has an attribution, particular values of $\boldsymbol{\lambda}$ exist such that the observability is lost.

Clearly, for a JF net, if the firing rates of the transitions are randomly chosen in $\mathbb{R}_{>0}$, the probability to obtain this cancelation is null and the already introduced weaker concept of observability appears: *generic observability*. This property is similar with the observability in *linear structured systems* [56] and can be studied using graph-based approaches. In fact, for self-loop free JF nets, the associated graph that is used in [43] to characterize the generic observability of linear structured systems can be obtained from the Petri net structure just removing all transitions, adding an arc from p_i to p_j if $p_j \in (p_i{}^\bullet)^\bullet$ and adding a self-loop to each place [124]. If at least one place from each terminal strongly connected p-component is measured, all states of the associated graph are output connected and the net system is *generic observable*.

Concluding, for JF TCPN, i.e., the net subclass that defines linear systems, all three concepts of observability can be easily characterized: (i) using the theory of linear systems for standard observability, i.e., the observability matrix should have full rank (where the observability matrix of a system with dynamic matrix $\boldsymbol{A}_i$ is $\vartheta_i = [\boldsymbol{S}^T \ (\boldsymbol{S}\boldsymbol{A}_i)^T \ \ldots \ (\boldsymbol{S}\boldsymbol{A}_i^{|P|-1})^T]^T$ [118]); (ii) the graph based results in [43] for generic observability, i.e., for self-loop free nets, measuring one place from each terminal strongly-connected p-componet; (iii) using graph based approach for structural observability of AF nets, i.e., measuring one place from each terminal strongly connected p-component, and, eventually, combining graph approaches with algebraic manipulations for nets with attributions.

Finally, let us introduce *joins* (otherwise stated, *rendez-vous*). According to the flow definition, in this case the system is not linear. Let us concentrate on infinitesimal time observability. Obviously, observability of all linear systems (or operation modes) is a necessary condition for the observability, but unfortunately it is not enough. It may happen that the continuous state estimation fits with different discrete states, i.e., observing some places, it may happen that more than one linear system satisfies the observation. For example, let us consider the TCPN in fig. 13(b) and p_3 the only measured place. This system has two modes corresponding to two linear systems. Let us interpret why it is not observable using the previous graph approach. Assume that the first mode is such that $f_3(\tau) = \lambda_3 \cdot m_1$. If we compute the backward path from p_1 to p_3 it is as if we ignore the arc (p_2, t_3). It is straightforward to see that $p_3 t_3 p_1 t_2 p_2$ is obtained. On the other hand, for the second mode, the arc (p_1, t_3) is ignored and the obtained path is: $p_3 t_3 p_2 t_1 p_1$. Obviously, if $\lambda_1 = \lambda_2$, the same set of differential equations is obtained and will be impossible to distinguish between two states [124]. For example, taking $\boldsymbol{m}_1 = [1\ 2\ 0]^T \in \mathcal{R}_1 \setminus \mathcal{R}_2$ and $\boldsymbol{m}_2 = [2\ 1\ 0]^T \in \mathcal{R}_2 \setminus \mathcal{R}_1$ both have the same observations, what correspond to $\vartheta_1 \cdot \boldsymbol{m}_1 = \vartheta_2 \cdot \boldsymbol{m}_2$, were ϑ_i are the observability matrices of the two operation modes.

Definition 16 *Two operation modes* 1 *and* 2 *of a TCPN system are* distinguishable *if for any* $\boldsymbol{m}_1 \in \mathcal{R}_1 \setminus \mathcal{R}_2$ *and any* $\boldsymbol{m}_2 \in \mathcal{R}_2 \setminus \mathcal{R}_1$ *the observation* $\boldsymbol{y}_1(\tau)$ *for the trajectory through* $\boldsymbol{m}_1$ *and the observation* $\boldsymbol{y}_2(\tau)$ *for the trajectory through* $\boldsymbol{m}_2$ *are* different *on the interval* $[0, \epsilon)$ *for all* $\epsilon > 0$.

If all pairs of modes are distinguishable, it is always possible to uniquely assign an *operation mode* (corresponding to a configuration, also defining a region) to an observed continuous state. Assuming that a pair of modes are observable, a LPP can be proposed to check their distinguishability. Unfortunately, this LPP may suffer from numerical problems since we have to find interior points of some regions and it is well known that strict inequalities are problematic to be implemented. Let us consider the following quadratic programming problem (QPP):

$$
\begin{array}{ll}
z = \max & \boldsymbol{\beta}^T \cdot \boldsymbol{\beta} \\
\text{s.t.} & \vartheta_1 \cdot \boldsymbol{m}_1 - \vartheta_2 \cdot \boldsymbol{m}_2 = 0 \\
 & \boldsymbol{\beta} = \boldsymbol{m}_1 - \boldsymbol{m}_2 \\
 & \boldsymbol{m}_1 \in \mathcal{R}_1 \\
 & \boldsymbol{m}_2 \in \mathcal{R}_2
\end{array}
\tag{14}
$$

First, let us observe that if the feasible set of (14) is empty, operation modes are distinguishable. If in QPP (14) $z = 0$, using the fact that both systems are observable, i.e., ϑ_1 and ϑ_2 have both full rank, $\boldsymbol{m}_1 = \boldsymbol{m}_2$ is obtained. Therefore, there exist no interior markings $\boldsymbol{m}_1 \in \mathcal{R}_1$ and $\boldsymbol{m}_2 \in \mathcal{R}_2$ with the same observation, i.e., $\vartheta_1 \cdot \boldsymbol{m}_1 = \vartheta_2 \cdot \boldsymbol{m}_2$, and the modes are distinguishable. Finally, if the solution is $z > 0$ we cannot say nothing about distinguishability of the modes. In this last case, for a particular solution of (14) a small variation of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ can be considered by assuming constant flow during this small time interval. If $\vartheta_1 \cdot \Delta\boldsymbol{m}_1 = \vartheta_2 \cdot \Delta\boldsymbol{m}_2$ with $\Delta\boldsymbol{m}_1 = \boldsymbol{A}_1 \cdot \boldsymbol{m}_1$ and $\Delta\boldsymbol{m}_2 = \boldsymbol{A}_2 \cdot \boldsymbol{m}_2$ then the operations mode are undistinguishable. Moreover, the exact solution of (14) is not necessary to be computed and if a feasible solution with $z > \delta$, with δ a small positive number, is found the search can be stopped. An immediate criterium for observability is obtained [124]:

Proposition 17 *A timed continuous Petri net system $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ under infinite server semantics is observable in infinitesimal time iff:*

1. *All pairs of modes are distinguishable,*

2. *For each mode, i.e., in each region, the associated linear system is observable.*

Based on the previous observations, the checking of the observability of a net system with joins is not a trivial task. For this reason, some results have been proposed in order to "delete" the joins without affecting the observable space. After that, observability can be checked using only the observability matrix. This reduction can be done under some general conditions if the net system is AF or EQ [124]. Notice that in the case of AF nets, since the joins can be removed, the structural observability problem is the same as for nets with forks and joins: one place from each terminal strongly connected p-component should be measured.

A complementary observability problem is presented in [112]. For the discrete-time model and measuring some places, the problem is to estimate the firing flow (speed) of the transitions and not the marking of the other places. Since the flow of a transition is the product between its firing rate (constant value) and the enabling degree, in some cases, measuring places or transitions is equivalent. Anyhow, in order to compute the flow through joins it is necessary to measure all of its input places. Moreover, we may also have different markings that have the same firing flow.

5.3 Design of observers

JF nets lead to linear systems, for which, Luenberger's observers [119, 135] are frequently used. Such an observer for a PN with a single mode can be expressed as: $\dot{\tilde{\boldsymbol{m}}} = \boldsymbol{A} \cdot \tilde{\boldsymbol{m}} + \boldsymbol{K} \cdot (\boldsymbol{z} - \boldsymbol{S} \cdot \tilde{\boldsymbol{m}})$ where $\tilde{\boldsymbol{m}}$ is the marking estimation, $\boldsymbol{A}$ and $\boldsymbol{S}$ are the matrices defining the evolution of the marking of the system and its output in continuous time, $\boldsymbol{z}$ is the output of the system, and $\boldsymbol{K}$ is a design matrix of parameters.

At a particular time instant, a continuous PN evolves according to a given operation mode, i.e., linear system. Thus, an online estimation can be performed by designing one (Luenberger) linear observer per each potential mode of the PN (in a similar way to [95] for a class of piecewise linear systems) and selecting the one that accomplishes certain properties. The "goodness" of an estimate can be measured by means of a *residual* [13]. Let us use the 1-norm $||\cdot||_1$, which is defined as $||\boldsymbol{x}||_1 = |\boldsymbol{x}_1| + \ldots + |\boldsymbol{x}_n|$. The residual at a given instant, $r(\tau)$, is the distance between the output of the system and the output that the observer's estimate, $\tilde{\boldsymbol{m}}(\tau)$, yields, i.e., $r = ||\boldsymbol{S} \cdot \tilde{\boldsymbol{m}}(\tau) - \boldsymbol{z}(\tau)||_1$. In order to be *selectable*, the estimations of the observers must verify the following conditions:

- The residual must tend to zero.
- The estimations of the places in a synchronization have to be *coherent* with the operation mode for which they are computed.

Thus, at a given time instant, only coherent estimations are selectable. Moreover, a criterion must be established to decide which coherent estimation is, at a given time instant, the most appropriate. An adequate *heuristics* is to choose the coherent estimation with minimum residual.

Consider the TCPN system in fig. 14. Let its output be the marking of place p_1, i.e., $\boldsymbol{m}[p_1] = \boldsymbol{S} \cdot \boldsymbol{m}$, where $\boldsymbol{S} = [1\ 0\ 0]$. The net has two configurations:

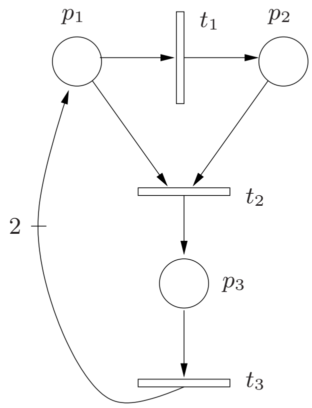


Figure 14: A net system with two operation modes. A deadlock is reachable by emptying the siphon $\{p_1, p_3\}$

$\mathcal{C}_1 = \{(p_1, t_1), (p_1, t_2), (p_3, t_3)\}$ and $\mathcal{C}_2 = \{(p_1, t_1), (p_2, t_2), (p_3, t_3)\}$. For the linear system corresponding to $\mathcal{C}_1$ m_2 is not observable. However, for the linear system corresponding to $\mathcal{C}_2$ the marking of all the places can be estimated. Let $\boldsymbol{\lambda} = [0.9\ 1\ 1]^T$ and $\boldsymbol{m}_0 = [3\ 0\ 0]^T$. The marking evolution of this system is depicted in fig. 15(a).

One observer per operation mode will be designed. Let the initial state of observer 1 be $\boldsymbol{e}_{01} = [1\ 2]^T$ and its eigenvalues be $[-12+2\cdot\sqrt{3}\cdot i,\ -12-2\cdot\sqrt{3}\cdot i]$. Since observer 1 can only estimate m_1 and m_3, the first component of its state vector corresponds to the estimation of m_1, and its second component to the estimation of m_3. For observer 2, let the initial state be $\boldsymbol{e}_{02} = [1\ 0\ 2]^T$ and its eigenvalues be $[-15,\ -12+2\cdot\sqrt{3}\cdot i,\ -12-2\cdot\sqrt{3}\cdot i]$. The evolution of the coherent estimation with minimum residual is shown in fig. 15(a).

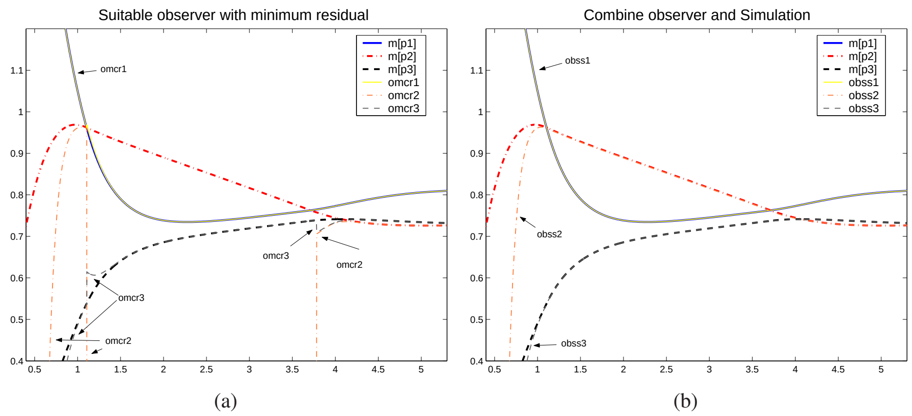


Figure 15: The marking evolution is given by $(\boldsymbol{m}[p_1],\ \boldsymbol{m}[p_2],\ \boldsymbol{m}[p_3])$. (a) The estimate of the minimum residual and coherent observer is $(omcr1,\ omcr2,\ omcr3)$. (b) The estimate of the observer that makes use of a simulation is $(obss1,\ obss2,\ obss3)$.

The resulting estimation can be improved by taking into account some considerations. When the first system switch happens, the estimation becomes discontinuous and, what is more undesirable, the estimation for the marking of p_3 becomes worse. A similar effect happens when the second system switch occurs. Another undesirable phenomenon is that, after the first switch, the estimation of m_2 just disappears (since it is unobservable in configuration $\mathcal{C}_1$).

One way to avoid discontinuities in the resulting estimation, is to use the estimation of the observer that is going to be filtered out in order to update the estimation of the observer that is not going to be filtered out. This estimation update must be done when a system switch is detected. In order not to lose the estimation of the marking of a place when it was "almost perfectly" estimated (recall the case of m_2 when the first switch happened) a simulation of the system can be launched. The initial marking of this simulation is the estimation of the system just before the observability of the marking is lost. Such a simulation can be seen as an estimation for those markings that are not observable by the observer being considered. The simulation should only be carried out when an estimation for all the places exists and the residual is not significant. Fig. 15(b) shows the evolution of the estimation obtained by this strategy.

One of the main advantages is that the residual does not increase sharply when the mode of the system changes. Another interesting feature is that the use of a simulation allows one to estimate the marking of places that in some modes are in principle not observable: in fig. 15(b) it can be seen that the marking of p_2 can be estimated, even when it is unobservable due to configuration $\mathcal{C}_1$ being active.

5.4 Observability and observers in other fluid models

Observability and state estimation problems in systems modeled by a continuous PN have been studied also in the case of *untimed* models or assuming a timed *finite* server semantics. Anyhow, the studied problems are a little bit different in both cases. Nevertheless, in both cases it is assumed that the initial marking is known (and not unknown as in previous section) and the set of transitions is partitioned in two: *observable* and *unobservable* transitions (hence transitions are observed and not places). Obviously, given a sequence of observed transitions, it is impossible to uniquely determine the actual marking, hence a set of markings will be the solution. This set is called the *set of consistent markings* and contains all markings in which the system may be given the actual observation.

In [125], *untimed* CPN are considered. When an observable transition fires, its firing quantity is measured and the problem is to obtain the set of consistent markings. It is proved that, under certain assumptions on the unobservable subnet, the set of consistent markings is convex. The main idea of an iterative algorithm to compute it is to start from each vertex of the previous set and compute the vertices of some polytopes. Taking the convex hull of all new vertices, the new set of consistent markings is obtained. The computational complexity of the algorithm is exponential because requires the computation of vertices, but the compact representation as a convex polytope is a real advantage.

Somehow related to the previous one, the problem of fault diagnosis is clearly a main issue in many engineering applications because of the practical need of ensuring the correct and safe functioning of systems. Using the characterization of the set of consistent markings and the algorithm to compute it, the problem of fault detection for systems modeled by untimed CPN has been recently addressed [145, 146]. Three diagnosis states have been considered: N - the fault has not occurred; U - the fault may have occurred or not (uncertain state); and F - the fault surely occurred. Given an observation, the diagnosis state is computed solving two LPPs. The main advantage of fluidification for fault diagnosis is that it enables to deal with more general Petri net structures than that considered in discrete approaches [24, 69]. In particular, the unobservable subnet needs not be acyclic.

In the case of TCPN under *finite server semantics*, the problem has been considered in [120]. Once again, it is assumed that the initial marking is known but no observation is available. Thus the observation problem reduces to determining the set of markings, in which the net may be at a given time. This problem is similar to that of time-reachability for continuous models. It is shown under which conditions the reachability set of the timed net under finite server semantics coincides with that of the untimed one. A procedure to compute the minimum time ensuring that the set of consistent markings is equal to the reachability set of untimed system is given for some net classes.

6 Controllability and control

Controllability is an important property in every kind of dynamic system. It is related to the capability of being driven in a certain desirable way. Continuous Petri nets are relaxations of discrete Petri nets, but at the same time, they are continuous-state systems (in fact, they are technically hybrid systems in which the discrete state is implicit in the continuous one). In this way, it seems natural to consider two different approaches for the controllability and control concepts: 1) at the discrete level, the extension of control techniques used in discrete PN's, such as the supervisory-control theory (for instance, [71, 85, 87]) and 2) at the fluidized level, the application of control techniques developed for continuous-state systems. Usually, the control objective in the first approach is to meet some safety specifications, like avoiding *forbidden* states, by means of disabling transitions at particular states. The objective of the second approach consists in driving the system, by means of a (usually) continuous control action, towards a desired steady state, or state trajectory (see, for instance, [35]). Regarding continuous Petri nets, most of the specific works that can be found in the literature deals with the second control approach applied to the *infinite* server semantics model.

Like in discrete PNs, the control action is applied through the transitions. This may only consist in the reduction of the flow, because transitions (machines for example) should not work faster than their nominal speed. A partition of the transitions set T is made, leading to sets of controllable (T_c) and uncontrollable (T_{nc}) transitions. The control vector $\boldsymbol{u} \in \mathbb{R}^{|T|}$ is defined s.t. u_i represents the control action on t_i. Assuming infinite server semantics, since u_i represents a reduction of the flow, then $0 \leq u_i \leq \lambda_i \cdot enab(t_i, \boldsymbol{m})$. The behavior of a *forced* (or controlled) continuous Petri net can be described by the state equation:

$$\begin{array}{c} \dot{\boldsymbol{m}} = \boldsymbol{C\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m} - \boldsymbol{Cu} \\ \text{subject to } \boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m} \text{ and } \forall t_i \in T_{nc},\, u_i = 0. \end{array} \tag{15}$$

Enforcing a desired target marking in a continuous PN is analogous to reaching an average marking in the original discrete model (assuming that the continuous model approximates the discrete one), which may be interesting in several kinds of systems. This idea has been illustrated by different authors. For instance, in [6] it was proposed a methodology for the control of open and closed manufacturing lines. The control actions consist in modifying the maximal firing speeds of the controlled transitions. It was also illustrated how the control law can be applied to the original discrete Petri net model (a T-timed model with constant firing delays). This approach has been used in [111] and [103] as well, in the same context of manufacturing lines. A related approach was presented in [166], for a stock-level control problem of an automotive assembly line system originally modeled as a stochastically timed discrete Petri net [59]. The resulting scheme allows to control the average value of the marking at the places that represent the stock-level, by means of applying additional delays to the controllable transitions.

For continuous PNs under infinite server semantics, controllability is considered in sections 6.1-6.3, while different control approaches are recalled in sections 6.4-6.6. Section 6.7 is devoted to a control technique for systems under finite server semantics.

6.1 On controllability: from the classical concept to bounded input controllability

The control objective considered here consists in driving the system, by applying a control law, towards a desired steady state, i.e., a *set-point* control problem, frequently addressed in continuous-state systems. This control objective is related to the classical controllability concept, according to which a system is controllable if for any two states $\boldsymbol{x}_1, \boldsymbol{x}_2$ of the state space it is possible to transfer the system from $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$ in finite time (see, for instance, [35]).

Few works have addressed the study of controllability in the context of continuous Petri nets. For instance, in [5] it is studied for linear nets (Join-Free nets), pointing out that the classical rank condition is not sufficient (detailed in subsection 6.3). In [91] the controllability was studied for Join-Free continuous nets from a different perspective, by characterizing the set of markings that can be reached and maintained. Nevertheless, those results are difficult to extend to general subclasses of nets, where the existence of several regions makes the general reachability problem untractable.

Despite those results, in [122] it was pointed out that TCPN systems are frequently not controllable according to the classical controllability concept, due to the marking conservation laws imposed by P-flows. In detail, if $\boldsymbol{y}$ is a *P-flow* then any reachable marking $\boldsymbol{m}$ must fulfill $\boldsymbol{y}^T\boldsymbol{m} = \boldsymbol{y}^T\boldsymbol{m}_0$, defining thus a *state invariant*. Nevertheless, the study of controllability "over" this invariant is particularly interesting. This set is formally defined as $Class(\boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{R}^{|P|}_{\geq 0} | \boldsymbol{B}_y^T\boldsymbol{m} = \boldsymbol{B}_y^T\boldsymbol{m}_0\}$, where $\boldsymbol{B}_y$ is a basis of P-flows, i.e., $\boldsymbol{B}_y^T\boldsymbol{C} = \boldsymbol{0}$. For a general TCPN system, every reachable marking belongs to $Class(\boldsymbol{m}_0)$.

Another issue that appears in TCPN systems is the nonnegativeness and boundedness of the input, i.e., $\boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m}$. Considering these issues, an appropriate local controllability concept was proposed in [164]:

Definition 18 *The TCPN system* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ is controllable with bounded input *(BIC)* over $S \subseteq Class(\boldsymbol{m}_0)$ *if for any two markings* $\boldsymbol{m}_1, \boldsymbol{m}_2 \in S$ *there exists an input* $\boldsymbol{u}$ *transfering the system from* $\boldsymbol{m}_1$ *to* $\boldsymbol{m}_2$ *in finite or infinite time, and it is suitably bounded, i.e.,* $\boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m}$*, and* $\forall t_i \in T_{nc}\ u_i = 0$ *along the marking trajectory.*

6.2 Controllability if all the transitions are controllable: consistency

The controllability in continuous PNs, when all the transitions are controllable, depends only on the structure of the net. Intuition for this can be gained by rewriting the state equation as:

$$\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{w} \tag{16}$$

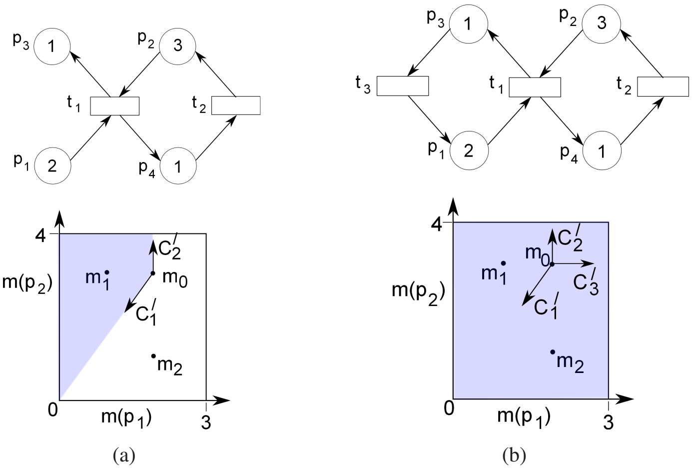


Figure 16: Two TCPN systems with identical P-flows. The shadowed areas correspond to the sets of reachable markings. Only the system (b) is consistent and controllable over $Class(\boldsymbol{m}_0)$.

where the innovation vector $\boldsymbol{w} = \boldsymbol{\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m} - \boldsymbol{u}$ can be seen as an auxiliary input. The constraints for $\boldsymbol{u}$ are transformed into $\boldsymbol{0} \leq \boldsymbol{w} \leq \boldsymbol{\Lambda\Pi}(\boldsymbol{m})\boldsymbol{m}$. In this way, given a marking $\boldsymbol{m}_1 \in Class(\boldsymbol{m}_0)$, if $\exists \boldsymbol{\sigma} \geq \boldsymbol{0}$ such that $\boldsymbol{C\sigma} = (\boldsymbol{m}_1 - \boldsymbol{m}_0)$ then $\boldsymbol{m}_1$ is reachable from $\boldsymbol{m}_0$. This can be achieved by setting $\boldsymbol{w} = \alpha\boldsymbol{\sigma}$ (with a small enough $\alpha > 0$), so the field vector results $\dot{\boldsymbol{m}} = \boldsymbol{C}\alpha\boldsymbol{\sigma} = \alpha(\boldsymbol{m}_1 - \boldsymbol{m}_0)$ which implies that the system will evolve towards $\boldsymbol{m}_1$ describing a straight trajectory (assuming that the required transitions can be fired from this marking, what always happens if $\boldsymbol{m}$ is a relative interior point of $Class(\boldsymbol{m}_0)$).

Consider for instance the TCPN of fig. 16(a) and the markings $\boldsymbol{m}_0 = [2\ 3\ 1\ 1]^T$, $\boldsymbol{m}_1 = [1\ 3\ 2\ 1]^T$ and $\boldsymbol{m}_2 = [2\ 1\ 1\ 3]^T$. Since this system has 2 P-semiflows (involving $\{p_1, p_3\}$ and $\{p_2, p_4\}$ respectively), the marking of two places is sufficient to represent the whole state. For this system $\exists \boldsymbol{\sigma} \geq \boldsymbol{0}$ such that $\boldsymbol{C\sigma} = (\boldsymbol{m}_1 - \boldsymbol{m}_0)$, but $\nexists \boldsymbol{\sigma} \geq \boldsymbol{0}$ such that $\boldsymbol{C\sigma} = (\boldsymbol{m}_2 - \boldsymbol{m}_0)$, so, $\boldsymbol{m}_1$ is reachable but $\boldsymbol{m}_2$ is not. The shadowed area in fig. 16(a) corresponds to the set of reachable markings, note that it is the convex cone defined by vectors $\boldsymbol{c}'_1$ and $\boldsymbol{c}'_2$, which represent the columns of $\boldsymbol{C}$ (here restricted to p_1 and p_2).

This structural reachability reasoning leads to a simple and full characterization of controllability [164]:

Proposition 19 *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system in which all the transitions are controllable. The system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ is BIC over the interior of $Class(\boldsymbol{m}_0)$ iff the net is* consistent. *Furthermore, the controllability is extended to the whole $Class(\boldsymbol{m}_0)$ iff (additionally to consistency) there exist no* empty siphon *at any marking in $Class(\boldsymbol{m}_0)$.*

Conditions of propositions 3 (regarding lim-reachability) and 19 are equivalent (the non existence of *empty siphons* is equivalent to the fireability of all the transitions). Note that the controllability does not depend on the timing $\boldsymbol{\lambda}$.

The key condition here is consistency, i.e., $\exists \boldsymbol{x} > 0$ such that $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$. Remember that a reachable marking $\boldsymbol{m} \geq \boldsymbol{0}$ fulfills $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$ with $\boldsymbol{\sigma} \geq \boldsymbol{0}$, which implies $\boldsymbol{B}_y^T\boldsymbol{m} = \boldsymbol{B}_y^T\boldsymbol{m}_0$ (equivalently, $\boldsymbol{m} \in Class(\boldsymbol{m}_0)$). On the opposite sense, if the net is consistent then $\forall \boldsymbol{m} \geq \boldsymbol{0}$ s.t. $\boldsymbol{B}_y^T \boldsymbol{m} = \boldsymbol{B}_y^T \boldsymbol{m}_0$ (i.e., $\boldsymbol{m} \in Class(\boldsymbol{m}_0)$) it exists $\boldsymbol{\sigma} \geq \boldsymbol{0}$ s.t. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, thus $\boldsymbol{m}$ is reachable (assuming $\boldsymbol{\sigma}$ is fireable). A very informal and intuitive explanation is that consistency permits movements of marking in any direction inside the reachability space (see fig. 16(b)), i.e., if there exists $\boldsymbol{\sigma}$ such that $\boldsymbol{m}_1 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, under consistency any $\boldsymbol{\sigma}' = \boldsymbol{\sigma} + k \cdot \boldsymbol{x} \geq 0$, permits the reachability of $\boldsymbol{m}_1$.

Consider again the TCPN system of fig. 16(a). By using the proposition 19 it can be verified that this TCPN is not controllable over $Class(\boldsymbol{m}_0)$, because the net is not consistent. Now, consider the system of fig. 16(b). In this case, for any marking $\boldsymbol{m} \in Class(\boldsymbol{m}_0)$, the vector $(\boldsymbol{m} - \boldsymbol{m}_0)$ is in the convex cone defined by the vectors $\boldsymbol{c}_1'$, $\boldsymbol{c}_2'$ and $\boldsymbol{c}_3'$; which occurs due to the consistency of the net and implies that $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$). Moreover, since at the border markings of $Class(\boldsymbol{m}_0)$ there are not unmarked siphons then, according to the proposition 19, the system is BIC over $Class(\boldsymbol{m}_0)$.

6.3 Controllability with uncontrollable transitions over stationary states

Systems with uncontrollable transitions are not controllable over $Class(\boldsymbol{m}_0)$, even for consistent nets. In this case, a smaller set of markings need to be considered. This idea was explored in [91], where a set named Controllability Space (CS), over which the system is controllable, was characterized for Join-Free nets. Nevertheless, this set depends on the marking, thus its characterization for general subclasses of nets is difficult. The existence of several regions makes the general reachability problem untractable. For practical reasons, the controllability was studied in [164] over sets of *equilibrium markings* : $\boldsymbol{m}^q \in Class(\boldsymbol{m}_0)$ is an equilibrium marking if $\exists \boldsymbol{u}^q$ suitable such that $\boldsymbol{C}(\boldsymbol{\Lambda\Pi}(\boldsymbol{m}^q)\boldsymbol{m}^q - \boldsymbol{u}^q) = \boldsymbol{0}$. They represent the *possible stationary operating points* of the original discrete system. These markings are particularly interesting, since controllers are frequently designed in order to drive the system towards a desired *stationary operating point*.

Since inside each region $\mathcal{R}_i$ (defined in subsection 4.3) the state equation is linear ($\boldsymbol{\Pi}(\boldsymbol{m})$ is constant), it becomes convenient to study, in a first step, the controllability over equilibrium markings in each region and later over the union of them. This approach is supported by the following proposition:

Proposition 20 *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system. Consider some equilibrium sets S_1, S_2,..., S_j related to different regions $\mathcal{R}_1$, $\mathcal{R}_2$,..., $\mathcal{R}_j$. If the system is BIC (in finite time) over each one and their union $\bigcup_{i=1}^{j} S_i$ is connected, the system is BIC over the union.*

The connectivity of the set of all the equilibrium markings in $Class(\boldsymbol{m}_0)$ has not been demonstrated for the general case. Nevertheless, in every studied system such property holds.

As an example, consider the timed continuous marked graph of fig. 17 with $T_c = \{t_4\}$ and $\boldsymbol{\lambda} = [1\,1\,1\,2]^T$. There are four possible configurations according to the structure, but given the initial marking, one of them cannot occur. The polytope in fig. 17 represents the $Class\{\boldsymbol{m}_0\}$. Since the system has 3 P-semiflows, the marking at $\{p_1, p_3, p_5\}$ is enough to represent the whole state. This is divided into the regions $\mathcal{R}_1$, $\mathcal{R}_3$ and $\mathcal{R}_4$, related to the feasible configurations. The segments $E_1 = [\boldsymbol{m}_1, \boldsymbol{m}_2]$, $E_3 = [\boldsymbol{m}_2, \boldsymbol{m}_3]$ and $E_4 = [\boldsymbol{m}_3, \boldsymbol{m}_4]$ are the sets of equilibrium markings in regions

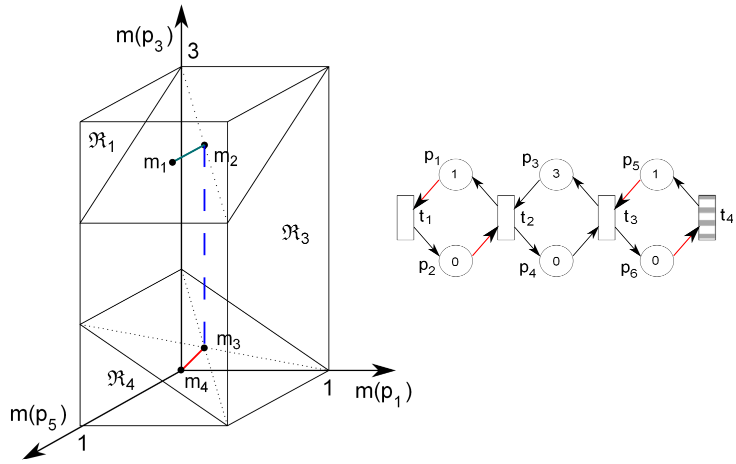


Figure 17: TCPN system with its $\mathbb{E}$. Transition t_4 is the only controllable one. There are four possible configurations: $\mathcal{C}_1 = \{(p_2,t_2), (p_4,t_3)\}$, $\mathcal{C}_2 = \{(p_3,t_2), (p_4,t_3)\}$, $\mathcal{C}_3 = \{(p_2,t_2), (p_5,t_3)\}$ and $\mathcal{C}_4 = \{(p_3,t_2), (p_5,t_3)\}$, however, $\mathcal{C}_2$ cannot occur from the given $\boldsymbol{m}_0$ because p_3 and p_4 cannot concurrently constrain t_2 and t_3, respectively. Equilibrium sets depend on the timing, but regions do not.

$\mathcal{R}_1$, $\mathcal{R}_3$ and $\mathcal{R}_4$, respectively. Since the union of E_1, E_3 and E_4 is connected, if the system were BIC over each E_i (this will be explored in a forthcoming example) then, according to Proposition 20, the system would be BIC over $E_1 \cup E_3 \cup E_4$. For instance, the system could be driven from $\boldsymbol{m}_3$ to $\boldsymbol{m}_1$ and in the opposite sense.

In a given region $\mathcal{R}_i$ the TCPN system is linear and time-invariant, then some of the classical results in control theory can be used for its analysis. Null-controllability (controllability around the origin) of this kind of systems with input constraints was studied in [21]. Recalling from there, if a linear system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, with input constraint $\boldsymbol{u} \in \Omega$ (called the set of admissible inputs), is controllable then the controllability matrix $Contr(\boldsymbol{A}, \boldsymbol{B}) = [\boldsymbol{B}\, \boldsymbol{AB} \ldots \boldsymbol{A}^{n-1}\boldsymbol{B}]$ has full rank (equivalently, $\forall \boldsymbol{x}_1, \boldsymbol{x}_2$: $\exists \boldsymbol{z}$ s.t. $(\boldsymbol{x}_2 - \boldsymbol{x}_1) = Contr(\boldsymbol{A}, \boldsymbol{B}) \cdot \boldsymbol{z}$). Furthermore, if $\mathbf{0}$ is in the interior of Ω then the previous rank condition is also sufficient for null-controllability. Otherwise, if there are inputs that can be only settled as positive (or negative) then the controllability depends also on the eigen-structure of the state matrix. These results can be adapted to TCPNs. For this, the state equation of a TCPN is firstly transformed in order to represent the behavior around an equilibrium marking $\boldsymbol{m}^q$, i.e., the evolution of $\Delta \boldsymbol{m} = \boldsymbol{m} - \boldsymbol{m}^q$. As a consequence, some transformed inputs $\Delta \boldsymbol{u} = (\boldsymbol{u} - \boldsymbol{u}^q)$ can be settled only as nonnegative while others can be settled as either positive or negative. The set of transitions related to this last kind of inputs is denoted as $T^i_{cf} \subseteq T_c$. Let us denote as E^*_i the set of all equilibrium markings in a region $\mathcal{R}_i$ s.t. $\Delta \boldsymbol{u}[T^i_{cf}]$ can be settled as either positive or negative (equivalently, $[\boldsymbol{\Lambda \Pi}_i \boldsymbol{m}^q]_j > u^q_j > 0$ for all $t_j \in T^i_{cf}$). In this way, it can be proved that if a TCPN is controllable over a set E^*_i then $\forall \boldsymbol{m}_2, \boldsymbol{m}_1 \in E^*_i$: $\exists \boldsymbol{z}$ s.t. $(\boldsymbol{m}_2 - \boldsymbol{m}_1) = Contr((\boldsymbol{C\Lambda\Pi}_i), \boldsymbol{C}[T_c]) \cdot \boldsymbol{z}$. This condition is only necessary, as already pointed out in [5], because the existence of input constraints. Furthermore, a system is controllable (in finite time) over E^*_i if $\forall \boldsymbol{m}_2, \boldsymbol{m}_1 \in E^*_i$: $\exists \boldsymbol{z}$ s.t. $(\boldsymbol{m}_2 - \boldsymbol{m}_1) = Contr((\boldsymbol{C\Lambda\Pi}_i), \boldsymbol{C}[T^i_{cf}]) \cdot \boldsymbol{z}$. This sufficient condition is also necessary if $T^i_{cf} = T_c$ (but not if $T^i_{cf} \subset T_c$). Note that now the controllability depends not only on the structure of the net, but also on the timing [164].

For instance, consider the region $\mathcal{R}_3$ in the system of fig. 17. For this, $T^3_{cf} = \{t_4\}$. Since $T^3_{cf} = T_c$ then the span condition introduced above is sufficient and necessary for controllability. In this case, it can be verified that the system is BIC over $E^*_3 = E_3$. Consider now the same system but with $\lambda_4 = 1$ instead of $\lambda_4 = 2$. In this case, $T^3_{cf} = \emptyset$ (this set depends on the timing), then we cannot use the same sufficient condition. Nevertheless, it is still fulfilled that $\forall \boldsymbol{m}_2, \boldsymbol{m}_1 \in E^*_3 : \exists \boldsymbol{z}$ s.t. $(\boldsymbol{m}_2 - \boldsymbol{m}_1) = Contr((\boldsymbol{C\Lambda\Pi}_3), \boldsymbol{C}[T_c]) \cdot \boldsymbol{z}$. Therefore, the controllability matrices do not provide enough information for deciding whether the system is BIC or not over E^*_3. By using another results from [164], it can be proved that the system is not BIC with $\lambda_4 = 1$, what leads to the conclusion that the controllability is a timing-dependent property.

6.4 Control when all the transitions are controllable

Through the following paragraphs and subsection 6.5, a few control techniques, proposed in the literature for continuous Petri nets, will be recalled. Similar to the *set-point* control problem in state-continuous systems, the control objective here consists in driving the system towards a desired target marking (a steady state, here denoted as $\boldsymbol{m}_d$). This desired marking can be selected, in a preliminarily planning stage, according to some optimality criterion [155], e.g., maximizing the flow as in subsection 4.3. Most of the work done on this issue is devoted to centralized dynamic control assuming that *all* the transitions are controllable. We will firstly present those control techniques that require to control all the transitions, while a couple of techniques (gradient-base and pole assignment), where uncontrollable transitions are considered, will be presented in the following subsection.

Fuzzy control [79]. The authors showed that the flow of a fluid transition, under infinite server semantics with an implicit self-loop, can be represented as the output of two fuzzy rules under the Sugeno model. It was proved that if the integral of the output of each fuzzy rule converges to a finite value then the resulting global fuzzy system (that represents the controlled flow) converges as well. Moreover, upper and lower bounds of this convergence were derived. Based on that, a *proportional fuzzy control* was proposed, proving convergence of the system to the desired output (the marking of a place $p_j \in P$, i.e., $\boldsymbol{m}_d[p_j]$), assuming that this is smaller than the initial upstream marking, i.e., $\boldsymbol{m}_d[p_j] \leq \boldsymbol{m}_0[p_i], \forall p_i \in {}^\bullet p_j$, which is not the general case.

Control for a piecewise-straight marking trajectory. Dealing with the tracking control problem of a mixed ramp-step reference signal, this approach was firstly explored in [93] for Join-Free nets and extended to general PNs in [94]. There, a high & low gain proportional controller is synthesized, while a ramp-step reference trajectory, as a sort of *path-planning* problem at a higher level, is computed. Let us detail the more simple synthesis procedure introduced in [7]. Consider the line l connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_d$, and the markings in the intersection of l with the region's borders, denoted as $\boldsymbol{m}^1_c$, $\boldsymbol{m}^2_c$, ..., $\boldsymbol{m}^n_c$. Define $\boldsymbol{m}^0_c = \boldsymbol{m}_0$ and $\boldsymbol{m}^{n+1}_c = \boldsymbol{m}_d$. Then, $\forall k \in \{0, n\}$ compute τ_k by solving the linear programming problem (LPP):

$$\begin{array}{ll} min\, \tau_k & \\ s.t.: & \boldsymbol{m}^{i+1}_c = \boldsymbol{m}^i_c + \boldsymbol{C} \cdot \boldsymbol{x} \\ & \boldsymbol{0} \leq \boldsymbol{x}_j \leq \boldsymbol{\lambda}_j \boldsymbol{\Pi}^z_{ji} min\{\boldsymbol{m}^i_{c,i}, \boldsymbol{m}^{i+1}_{c,i}\}\tau_k \\ & \forall j \in \{1, ..., |T|\} \text{ where i satisfies } \boldsymbol{\Pi}^z_{ji} \neq \boldsymbol{0} \end{array} \tag{17}$$

The control law to be applied is thus $\boldsymbol{w} = \boldsymbol{x}/\tau_k$ (the model is represented as in (16)), when the system is between the markings $\boldsymbol{m}^k_c$ and $\boldsymbol{m}^{k+1}_c$. The time required for reaching the desired marking is given by $\tau_f = \sum_{k=0}^{n} \tau_k$. *Feasibility* and *convergence* to $\boldsymbol{m}_f$ were proved in [7].

In order to obtain faster trajectories, intermediate states, not necessarily on the line connecting the initial and the target marking, can be introduced [94]. According to [7], they can be computed by means of a *bilinear* programming problem (BPP). The idea is to currently compute the intermediate markings $\boldsymbol{m}_c^k$, on the borders of the regions, that minimizes the total time $\tau_f = \sum_{k=0}^{n} \tau_k$ with some additional monotonicity constraints. Finally, the same algorithm can be adapted in order to recursively compute intermediate markings in the interior of the regions, obtaining thus faster trajectories.

Model predictive control (MPC) [121]. Here, two solutions were considered based on the *implicit* and *explicit* methods (see, for instance, [16]). The evolution of the timed continuous Petri net model (16), in *discrete-time*, is represented by the difference equation: $\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k)$, subject to the constraints $\boldsymbol{0} \leq \boldsymbol{w}(k) \leq \boldsymbol{f}(k)$ with $\boldsymbol{f}(k)$ being the flow without control, which is equivalent to $\boldsymbol{G} \cdot [\boldsymbol{w}^T(k), \boldsymbol{m}^T(k)]^T \leq \boldsymbol{0}$, for a particular matrix $\boldsymbol{G}$. The sampling Θ must be chosen small enough in order to avoid spurious markings, in particular, for ensuring the positiveness of the markings. For that, the following condition is required to be fulfilled $\forall\, p \in P \;:\; \sum_{t_j \in p^{\bullet}} \lambda_j \Theta < 1$.

By using this representation of the continuous PN, a MPC control scheme is derived in [121]. The goal is to drive the system towards a desired marking $\boldsymbol{m}_d$, while minimizing the quadratic performance index

$$\begin{aligned} J(\boldsymbol{m}(k), N) = (\boldsymbol{m}(k+N) - \boldsymbol{m}_d)' \boldsymbol{Z} (\boldsymbol{m}(k+N) - \boldsymbol{m}_d) \\ + \sum_{j=0}^{N-1} [(\boldsymbol{m}(k+j) - \boldsymbol{m}_d)' \boldsymbol{Q} (\boldsymbol{m}(k+j) - \boldsymbol{m}_d) \\ + (\boldsymbol{w}(k+j) - \boldsymbol{w}_d)' \boldsymbol{R} (\boldsymbol{w}(k+j) - \boldsymbol{w}_d)] \end{aligned}$$

where $\boldsymbol{Z}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite matrices and N is a given time horizon. This leads to the following optimization problem that needs to be solved in each time step:

$$\begin{array}{ll} \min J(\boldsymbol{m}(k), N) & \\ \text{s.t.} : \forall j \in \{0, ..., N-1\}, & \boldsymbol{m}(k+j+1) = \boldsymbol{m}(k+j) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k+j) \\ & \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}(k+j) \\ \boldsymbol{m}(k+j) \end{bmatrix} \leq \boldsymbol{0} \\ & \boldsymbol{w}(k+j) \geq \boldsymbol{0} \end{array} \tag{18}$$

In [121] it was shown that the standard techniques used for ensuring converge in linear/hybrid systems (i.e., terminal constraints or terminal cost) cannot be applied in continuous nets if the desired marking has zero components. Nevertheless, a particular control law, guaranteeing *asymptotic stability* for all possible final states, was proposed. Simulations showed that the horizon N is not required to be too large (actually, it is well known in classical systems theory that if $\exists \bar{N}$ s.t. $\forall \boldsymbol{m}_0$ and $\forall N \geq \bar{N}$, the finite and the infinite horizon controllers are equal). However, sometimes N is such that the computational time needed to solve the optimization problem becomes larger than the sampling period, making the implementation unfeasible.

An alternative MPC approach for this problem is the so-called *explicit* solution [16], where the set of all states that are controllable is split into polytopes. In each polytope the control command is defined as a piecewise affine function of the state. The closed-loop *stability* is guaranteed with this approach. On the contrary, when either the order of the system or the length of the prediction horizon are not small, the *complexity* of the explicit controller becomes quickly prohibitive. Furthermore, the computation of the polytopes sometimes is unfeasible.

Proportional control synthesis with LMI [102]. The proposed control scheme consists of a set of proportional (affine) control laws, one for each region. In detail, the controlled flow is represented, in discrete time, by $\boldsymbol{w}(k) = \boldsymbol{F}_r(\boldsymbol{m}(k) - \boldsymbol{m}_d) + \boldsymbol{R}$, where $\boldsymbol{R}$ is a vector and $\boldsymbol{F}_r$ is a gain matrix computed for each region (the subindex r denotes the $r-th$ region). In each region, the control and the marking are required to fulfill:

1. the input constraints: $\boldsymbol{0} \leq \boldsymbol{w}(k) \leq \boldsymbol{f}(k)$, where $\boldsymbol{f}(k)$ represents the flow without control,
2. the region membership: $\boldsymbol{m}(k) \in \mathcal{P}(\boldsymbol{G}_r, \boldsymbol{g}_r)$, where $\mathcal{P}(\boldsymbol{G}_r, \boldsymbol{g}_r) = \{\boldsymbol{m} | \boldsymbol{G}_r \boldsymbol{m} \leq \boldsymbol{g}_r\}$ is the inequality representation of the r-th region (a polyhedral),
3. the existence of a contractive invariant set (in order to prove closed-loop stability), which is stated as: $\boldsymbol{x}(k) \in \mathcal{P}(\boldsymbol{Q}, \boldsymbol{\mu}) \rightarrow \boldsymbol{x}(k+1) \in \mathcal{P}(\boldsymbol{Q}, \alpha\boldsymbol{\mu})$, where $\boldsymbol{x}(k) = (\boldsymbol{m}(k) - \boldsymbol{m}_d)$ is the current error, $\alpha < 1$ and $\mathcal{P}(\boldsymbol{Q}, \alpha\boldsymbol{\mu}) = \{\boldsymbol{x} | \boldsymbol{Q}\boldsymbol{x} \leq \alpha\boldsymbol{\mu}\}$ is the contractive set (so, the absolute error is monotonic decreasing).

The methodology consists in expressing the previous conditions as sets of linear matrix inequalities (LMI), one set for each region. The solution of a LMI can be achieved in polynomial time. Furthermore, *convergence* to the desired marking $\boldsymbol{m}_d$ is guaranteed. The main drawback of this approach is that a LMI must be solved for each region, but the number of these increases exponentially w.r.t. the number of synchronizations (joins).

ON-OFF (minimum-time) control for persistent nets [173]. Stronger results may be obtained when the problem is restricted to particular net subclasses. Accordingly, the minimum-time control problem was solved in this work for *persistent continuous Petri nets* (i.e., net systems where the enabling of any transition t_j cannot decrease by the firing of any other transition $t_i \neq t_j$). The solution is truly straightforward. First, a minimal firing count vector $\boldsymbol{\sigma}$ s.t. $\boldsymbol{m}_d = \boldsymbol{m}_0 + \boldsymbol{C}\boldsymbol{\sigma}$ is computed ($\boldsymbol{\sigma}$ is minimal if for any T-semiflow $\boldsymbol{x}$, $||\boldsymbol{x}|| \not\subseteq ||\boldsymbol{\sigma}||$, where $||\cdot||$ stands for the support of a vector). Later, the control law is defined, for each transition t_j, as:

$$\boldsymbol{u}[t_j] = \begin{cases} 0 & \text{if } \int_0^{\tau^-} \boldsymbol{w}[t_j] d\tau < \boldsymbol{\sigma}[t_j] \\ \boldsymbol{f}[t_j] & \text{if } \int_0^{\tau^-} \boldsymbol{w}[t_j] d\tau = \boldsymbol{\sigma}[t_j] \end{cases}$$

This means that if t_j has not been fired an amount of $\boldsymbol{\sigma}[t_j]$, then t_j is completely ON. Otherwise, t_j is completely OFF (it is blocked). It is proved that this ON-OFF control policy drives the system towards $\boldsymbol{m}_d$ in *minimum time*. An intuitive reason for this is that, for persistent nets, the firing order is irrelevant for reaching a marking. Thus, what only matters is the amount of firings required, which is provided by $\boldsymbol{\sigma}$.

Preliminary decentralized control techniques [8,170]. In order to deal with systems having large net structures (many places and transitions), it seems natural to consider decentralized and distributed control strategies. In a completely distributed approach, the model can be considered as composed of several subsystems that share information through *communication channels*, modeled by places. This problem has been addressed in [8] for a system composed of two MTS subsystems asynchronously connected through places. For each subsystem, a controller is designed. The mission of each local controller is to drive its corresponding subsystem from its initial marking to a required one, taking into account the interaction with the other subsystems. For this, it is required that neighboring local controllers share information, related to the possibility of concurrently reaching the target marking in every subsystem. A *consensus* algorithm is proposed for that task. Feasibility and concurrent convergence are demonstrated. In a second approach [170], the existence of an upper-level controller, named *coordinator*, is allowed. This coordinator may receive and send information to the local controllers, but it cannot apply control actions directly to the TCPN system. The existence of such coordinator increases the capability of the local controllers, allowing to consider wider classes for the net subsystems (they are assumed to be separately live and consistent, but they are not restricted to particular net subclasses). Affine control laws are proposed for local controllers. Feasibility and concurrent convergence to the required markings are proved.

6.5 Control with uncontrollable transitions

Gradient-base control with uncontrollable transitions [113]. Here, the input control actions consist in reducing the rates of the controllable transitions from their nominal maximum values, which is equivalent to reduce the flow, as considered along this section. Nevertheless, the goal of the control problem is slightly different, since it is no longer required to drive the whole marking of the system to a desired value, but only the marking of a subset of places (the *output* of the system). The analysis is achieved in discrete time. Let us provide the basic idea for the case of a single-output system. Firstly, a cost function is defined as $v(k) = 1/2\epsilon(k)^2$, where $\epsilon(k)$ denotes the output error. The control proposed has a structure like: $\boldsymbol{u}(k) = \boldsymbol{u}(k-1) - (\boldsymbol{s}(k)\boldsymbol{s}(k)^T + \alpha \boldsymbol{I})^{-1}\boldsymbol{s}(k)\epsilon(k)$, where the input $\boldsymbol{u}(k)$ is the rate of the controllable transitions and $\boldsymbol{s}(k)$ is the output sensitivity function vector with respect to the input (the gradient vector $\nabla_{\boldsymbol{u}}\, y$). The factor $\alpha > 0$ is a small term added to avoid ill conditioned matrix computations. The gradient is computed by using a first order approximation method. One of the advantages of this approach is that the change of regions (or configurations) is not explicitly taken into account during the computation of the gradient. Furthermore, a sufficient *condition for stability* is provided.

Pole assignment control with uncontrollable transitions [167]. In a first step, it is assumed that the initial and desired markings are equilibrium ones and belong to the same region. The control approach considered has the following structure: $\boldsymbol{u} = \boldsymbol{u}'_d + \boldsymbol{K}(\boldsymbol{m} - \boldsymbol{m}'_d)$, where $(\boldsymbol{m}'_d, \boldsymbol{u}'_d)$ is a suitable intermediate equilibrium marking. The gain matrix $\boldsymbol{K}$ is computed, by using any pole-assignment technique, in such a way that the controllable poles are settled as distinct, real and negative. Intermediate markings $\boldsymbol{m}'_d$, with their corresponding input $\boldsymbol{u}'_d$, are computed during the application of the control law (either at each sampling period or just at an arbitrary number of them) by using a given LPP with linear complexity that guarantees that the required input constraints are fulfilled. Later, those results are extended in order to consider several regions. For this, it is required that the initial and desired markings belong to a connected union of equilibrium sets, i.e., $\boldsymbol{m}_0 \in E_1^+$, $\boldsymbol{m}_d \in E_n^+$ and $\cup_{i=1}^n E_i^+$ is connected. Thus, there exist equilibrium markings $\boldsymbol{m}_1^q, ..., \boldsymbol{m}_{n-1}^q$ on the borders of consecutive regions, i.e., $\boldsymbol{m}_j^q \in E_j \cap E_{j+1}$, $\forall j \in \{1, .., j-1\}$. A gain matrix $\boldsymbol{K}_j$, satisfying the previously mentioned conditions, is computed for each region. Then, inside each jth region, the control action $\boldsymbol{u} = \boldsymbol{u}'_d + \boldsymbol{K}_j(\boldsymbol{m} - \boldsymbol{m}'_d)$ is applied, where $\boldsymbol{m}'_d$ is computed, belonging to the segment $[\boldsymbol{m}_j^q, \boldsymbol{m}_{j+1}^q]$, by using a similar LPP. It was proved that this control law can always be computed and applied (*feasibility*). Furthermore, *convergence* to the desired $\boldsymbol{m}_d$ was also demonstrated, whenever the conditions for controllability are fulfilled and $\cup_{i=1}^n E_i^+$ is connected (see section 6.3). The main drawback of this technique is that a gain matrix and a LPP have to be derived for each region in the marking path.

Table 4: Qualitative characteristics of control laws (assuming $\boldsymbol{m}_f > 0$). The following abbreviations are used: config. (configuration), min. (minimize), func. (function), exp. (exponential), compl. (complexity) and poly. (polynomial).

Technique	Computational issues	Optimality criterion	Subclass	T_{nc}	Stability
PW-straight trajectory	a LPP for each config.	heuristic for min. time	all	no	yes
MPC	poly. compl. on $\|T\|, N$	min. quadratic or linear func. of $\boldsymbol{m}, \boldsymbol{u}$	all	no	under suf. conditions
LMI	a LMI for each config.	none	all	no	yes
ON-OFF	linear compl. on $\|T\|$	minimum time	structurally persistent	no	yes
Gradient-based	poly. compl. on $\#$ outputs	min. quadratic error	all	yes	under a suf. condition
Pole-assignment	a pole-assignment for each config.	none	all	yes	yes

6.6 Preliminary comparison of control methods under infinite server semantics

Having several control methods available for timed continuous PNs, a question that may arise concerns the selection of the most appropriate technique for a given particular system and purpose. There are several properties that may be taken into account, like feasibility, closed-loop stability, robustness, computational complexity (for the synthesis and during the application), etc.

Table 4 summarizes a few qualitative properties of some of the control methods described above. Accordingly, provided a structurally persistent PN, the natural choice will be an $ON - OFF$ control law, since it does not exhibit computational problems, ensures convergence an provides the minimum-time transient behavior. For non-persistent nets, MPC ensures convergence and minimizes a quadratic criterion. Nevertheless, when the number of transitions grows, the complexity may become untractable. In such a case, control synthesis based on LMI or enforcing piecewise-straight trajectories would be more appropriated. Finally, control laws based on gradient-descendent and pole assignment methods are developed in order to deal with uncontrollable transitions. The synthesis of this last technique becomes tedious (but automatizable) when several configurations appear in the system, since a pole assignment is required for each configuration. This problem does not appear for the gradient based controller; on the contrary, this technique does not guarantee convergence for the general case, while the pole assignment does it.

Given a system with just few configurations and transitions, all of them being controllable, most of the described control laws could be synthesized and applied to it, ensuring convergence. In such case, the criterion for selecting one of them may be a quantitative one, like minimizing either a quadratic optimization criterion or the time spent for reaching the desired marking. At the present moment, such quantitative comparison has not been systematically made, but it is our intuition that the transient response of the mentioned techniques should be comparable (of the same order of magnitude), i.e., one technique could be the best for a TCPN system, while another technique could be better for a slightly different model.

6.7 Control under finite server semantics

In contrast to ISS, under finite server semantics (FSS), the flow vector, $\boldsymbol{f}$, is piecewise constant: it keeps constant until an event occurs, and events occur only when places become empty. Between events, the system is said to be at a *invariant behavior state* (IB - state) [49]. The concept of IB - state in FSS is similar to that of configuration in ISS. At an IB - state the flows of transitions are constant and therefore the markings of places increase or decrease linearly. Given a net $\mathcal{N}$ and a vector $\boldsymbol{\lambda}$, the flow vector under FSS just depends on the set of empty places. Hence, the number of potential IB - states is equal to the number of sets of places that can be simultaneously empty in the system. These differences entail that alternative techniques are required for the control of systems under FSS. We will focus on optimal control problems.

As in ISS, a transition t is *controllable* when its flow can be slowed down in a quantity that depends on the input, $\boldsymbol{u}[t]$, applied to it. The value $\boldsymbol{u}[t]$ is positive and upper limited by $\boldsymbol{\lambda}[t]$. The way of computing $\boldsymbol{f}$ is analogous to the one shown in section 4.1 for FSS, being now the maximum flow allowed by t, $\boldsymbol{\lambda}[t]-\boldsymbol{u}[t]$. Hence, if transition t is *strongly* enabled then $\boldsymbol{f}[t]=\boldsymbol{\lambda}[t]-\boldsymbol{u}[t]$. If t is *weakly* enabled $\boldsymbol{f}[t]$ will be computed considering $\boldsymbol{\lambda}[t]-\boldsymbol{u}[t]$ the upper bound for the flow of t. If t is neither strongly nor weakly enabled $\boldsymbol{f}[t]=0$. An alternative approach to control under FSS is developed in [12] where the firing speed of transitions is a control variable what allows to solve conflicts and optimization problems.

In the literature, there are several works dealing with optimal control in hybrid systems (remember that a CPN model under FSS is a particular class of hybrid system). Most of them can be roughly divided into two groups: those using continuous-time models (see, for example [32, 176]) and those using discrete-time models (see, for example [15]). Regarding continuous-time models, the main considered issues are the study of necessary trajectories to be optimal and the computation of optimal control laws by means of Hamilton-Jacobi-Bellman equations or the maximum principle. With respect to discrete-time models, a solution to optimal control problems was proposed in [15]. A drawback of time-discretization is that the length of the sampling period is not easy to define, since there often exists a trade-off between accuracy (short sampling period) and computational speed (long sampling period).

An intermediate approach between continuous and discrete time models consists of considering *event-driven models*. The evolution of an event-driven model is described in terms of *steps*, where steps are associated to the occurrence of events, i.e., to a place becoming empty in the case of continuous PNs. Each step contains the time instant at which the event occurred as well as the system state at that instant. This way, in contrast to usual discrete-time models, the time elapsed between events is not necessarily constant. Event-driven models (for instance, see [144] for models based on max-plus algebra) offer two interesting advantages: 1) Event-discretization does not imply loss of accuracy: The marking evolution of a continuous PN under FSS 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). The task of solving optimal control problems by using event-driven models of continuous nets is greatly eased if the net system is transformed to a Mixed Logical Dynamical (MLD) system [15]. The basic steps to transform a continuous PN into a MLD system are the following [96]:

1. Identify the potential IB - states of the continuous PN.
2. Describe the behavior of the PN under each IB - state.
3. Force that at least one place becomes empty at the occurrence of the next event.

Once a MLD system is obtained, it can be equipped with the objective function that is desired to be optimized. This produces a Mixed Integer Linear Programming Problem (MILP) whose objective function represents the pursued control goal, e.g., minimum-time, minimum-effort, minimum-displacement, etc, and whose solution contains the control actions that optimize the objective function [25]. Let us exemplify the event-driven control through the net in fig. 14 with $\boldsymbol{\lambda} = [1.5\,1\,2]^T$ and $\boldsymbol{m}_0 = [6\,0\,0]^T$. If no control action is applied to the system, it reaches a steady state marking with null throughput, i.e., a deadlock marking. An interesting control problem for such non live systems and for many manufacturing systems is to find control actions that maximize the throughput in the steady state, thus deadlocks are avoided. Such a control goal can be achieved by defining a MILP [96]. Notice that the considered system is bounded and has a unique T-semiflow $\boldsymbol{\chi} = 1$. Hence, maximizing the throughput of one transition in the steady state is equivalent to maximizing the throughput of any of the three transitions. Let us assume that t_3 is the only controllable transition. The obtained control is $\boldsymbol{u}[t_3] = 2$ during 2.4 time units, which leads to $\boldsymbol{m} = [0\,1.2\,2.4]^T$. At $\boldsymbol{m}$ the flow of all transitions is equal to 1. The control action required to keep $\boldsymbol{m}$ constant is $\boldsymbol{u}[t_3] = 1$.

7 Some concluding remarks and open issues

The fluidization of discrete event dynamic models is a classical technique used from the late sixties of the past century in the Queueing Network framework. As in Petri Nets, Forrester Diagrams or Stochastic Flow Models, happens to be particularly interesting for systems following production/consumption patterns. The material presented here represents the warp of an approximation, where three different weaves are essential: the autonomous or fully non-deterministic view, rooted in Theoretical Computer Science; the performance evaluation of unforced (or uncontrolled) models, strongly related to classical Operations Research perspectives; and observation, control, and related problems, central to Control Theory. Not surprisingly, all three disciplines play a major role in the arena of the formal study of DEDS. The explicit consideration of the relationships of the fluid model and the underlying discrete event one is also a crucial concern. For these reasons, aspects such as the *legitimacy* of the fluid relaxations or their *improvements* (through the elimination of spurious markings, the addition of noise to reflect the stochastic variability, or the convenience of the introduction of special server semantics) are important.

Fluidization means a loss of fidelity with respect to the discrete model, but as shown in this work, among the benefits is the substantial reduction in complexity of several important computational problems. For example, *convexity* is a property that, in general, makes optimization problems easier. Under the consistency of the net and the non-existence of empty siphons at the initial marking (that can be proved through LPPs), the state-space is *linearly* described, which does not mean that the behaviors that can be described are linear! Moreover, the ability to fire in isolation minimal T-semiflows causes behavioral and structural *synchronic relations* to collapse (thus, for example, the bound of a place can be straightforwardly computed in polynomial time). Under the above conditions, the analysis of reachability in untimed models, among other properties, is also solvable in polynomial time. Nevertheless, the introduction of time makes things less simple. For example, under infinite server semantics, continuous Petri Nets may simulate Turing Machines! Of course, this means that the theoretical *expressive power* is very satisfactory, but now several properties (such as the existence of steady state) are *undecidable*. Dealing with synthesis problems for timed models, and assuming that a steady state exists (for example when using the final-value theorem with the Laplace transform), the computation of the minimal marking for certain required cycle times or some optimal steady state control actions are also polynomial time problems. From a different perspective, the bridge to continuous control theory is also very challenging; nevertheless, PN structure theory based is the fact that if all transitions are controllable, controllability can also be decided in polynomial time. Obviously, the over-approximation that is fluidization entails that something is lost, but usually (particularly if the system is performance monotonic) the bigger the initial marking, the smaller the errors produced by the relaxation.

Another benefit of fluidization is that the fluid model permits the use of *infinitesimal perturbation analysis* providing algorithms for the gradients (derivatives) of sample performance functions [31]. This technique has been applied mainly to queueing models, where performance metrics of delay and throughput are of primary interest. Recently, the application of infinitesimal perturbation analysis to TCPN has been proposed in [175] and [72] where the gradient of the throughput with respect to fluid flow parameters is studied for marked graphs.

Even if the reader may be under the impression that there is a substantial amount of accumulated knowledge available, we would like to point out that there are plenty of fundamental issues that still require significant research efforts. Just to mention a few examples:

- When is it reasonable to fluidize a discrete event dynamic (both as untimed and as timed) model?
- Which is the most appropriate timing interpretation in order to approximate well the underlying discrete model? (Otherwise stated: which is the best server semantics for the problem under consideration?)
- Expressiveness of CPNs under infinite server semantics is high to the level of being able to simulate Turing machines. But this this is also means the existence of undecidabilities. In which net subclasses may we have simultaneously significant modeling and decidability power?
- For the different server semantics, what about the establishment of duality theories between observation and control? Can the interleaving of graph and algebraic based techniques be made more symmetric (not only as a sign of beauty and elegance, but also for practical purposes)?
- What about the efficient computation of sensitivity and optimization issues in continuous models (IPA techniques and others)?
- Observability and controllability criteria for distributed systems are needed.
- How to design more efficient observers, with fewer elements?

- What about the adequacy of continuous control laws for the underlying discrete event dynamic model? What are the limitations?

- To what extent, the theory of discrete and continuous models can help in the building of a more solid theory of hybrid systems? To what extent, the present theories of hybrid systems can help to improve the understanding of these fluid models, as already pointed out hybrid in a technical sense?

Of course, to most of the previous questions, elements of answers are already available, but there is a long way to cover, surely a beautiful travel to do in company.

Acknowledgement

This work benefits from the comments of the following colleagues and friends: R. David (INP de Grenoble), A. Giua (Univ. di Cagliari), E. Jiménez (Univ. de La Rioja), A. Ramírez (Cinvestad, Guadalajara, Mx) and J. Zaytoon (Univ. de Reims-Champagne-Ardennes). All of them and the anonymous reviewers are gratefully acknowledged. This work has been partially supported by the European Community's Seventh Framework Program under project DISC (Grant Agreement n. INFSO-ICT-224498), by CICYT - FEDER grants DPI2010-20413 and by Fundación Aragón I+D.

References

[1] M. Ajmone-Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Trans. on Software Engineering*, 15(7):832–846, 1989.

[2] H. Alla and R. David. Continuous and Hybrid Petri Nets. *Journal of Circuits, Systems, and Computers*, 8(1):159–188, 1998.

[3] E. Altman, T. Jimï¿$\frac{1}{2}$nez, and G. Koole. On the comparison of Queueing Systems with their Fluid limits. *Probability in the Engineering and Informational Sciences*, 15:165–178, 2001.

[4] R. Alur, C. Coucoubetis, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[5] A. Amer-Yahiaj, N. Zerhouni, A. El-Moudni, and M. Ferney. State variable description and controllability of a class of continuous Petri nets. In *Proc. of the IEEE Int. Symp. on Circuits and Systems*, 1996.

[6] A. Amrah, N. Zerhouni, and A. El-Moudni. On the control of manufacturing lines modelled by controlled continuous Petri nets. *Int. Journal of Systems Science*, 29(2):127–137, 1998.

[7] H. Apaydin-Ozkan, J. Julvez, C. Mahulea, and M. Silva. An efficient heuristics for minimum time control of continuous Petri nets. In *3rd IFAC Conf. on Analysis and Design of Hybrid Systems, ADHS09*, 2009.

[8] H. Apaydin-Ozkan, J. Julvez, C. Mahulea, and M. Silva. A control method for timed distributed continuous Petri nets. In *Proc. of the American Control Conf. 2010*, Baltimore, Maryland, USA, 2010. IEEE Press.

[9] F. Avram. Optimal control of fluid limits of queuing networks and stochasticity corrections. In *Mathematics of stochastic manufacturing systems*, volume 33, pages 1–37. Amer Mathematical Society, 1997.

[10] G. Balbo and M. Silva, editors. *Performance Models for Discrete Event Systems with Synchronozations: Formalisms and Analysis Techniques, Proc. of Human Capital and Mobility MATCH Performance Advanced School*, Jaca, Spain, 1998. Available online: http://webdiis.unizar.es/GISED/?q=news/matchbook.

[11] 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.

[12] 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.

[13] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. L. Sangiovanni-Vincentelli. Design of Observers for Hybrid Systems. In Claire J. Tomlin and Mark R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, pages 76–89. Springer-Verlag, Berlin Heidelberg New York, 2002.

[14] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *Automatic Control, IEEE Trans. on*, 45(10):1864–1876, 2002.

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

[16] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[17] L. Benevenuti and L. Farina. Positive and compartmental systems. *IEEE Trans. on Automatic Control*, 47(2):370–373, Feb 2002.

[18] J.A. Bergstra and J.W. Klop. Fixed point semantics in process algebra. Technical report iw 208, Mathematical Centre, Amsterdam, 1982.

[19] D. Bertsimas, D. Gamarnik, and J.N. Tsitsiklis. Stability conditions for Multiclass Fluid Queueing Networks. *IEEE Trans. On Automatic Control*, 41(11):1618–1631, 1996.

[20] G. Bolch, S. Greiner, H. de Meer, and K. Trivedi. *Queueing Networks and Markov Chains*. Wiley, New York, 1998.

[21] R. Brammer. Controllability in linear autonomous systems with positive controllers. *SIAM Journal Control*, 10(2):329–353, 1972.

[22] G. W. Brams. *Réseaux de Petri: Théorie et Pratique*. Masson, 1983.

[23] M. Bramson. *Stability of Queueing Networks*. Number 1950 in Lecture Notes in Mathematics. Springer, New York, 2008 (also published in: Probability Surveys, 5:169-345, 2008).

[24] M.P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9):1531–1539, 2010.

[25] S. Di Cairano, A. Bemporad, and J. Júlvez. Event-driven optimization-based control of hybrid systems with integral continuous-time dynamics. *Automatica*, 45(5):1243 – 1251, 2009.

[26] J. Campos, JM Colom, and M. Silva. Improving throughput upper bounds for net based models of manufacturing systems. *Robotics and Flexible Manufacturing Systems*, pages 281–294, 1992.

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

[28] L. Cardelli. On process rate semantics. *Theoretical Computer Science*, 391(3):190–215, 2008.

[29] J. Cardoso and H. Camargo (eds.). *Fuzziness in Petri Nets*, volume 22 of *Studies in Fuzziness and Soft Computing*. Springer–Verlag, 1999.

[30] C. G. Cassandras. Stochastic Flow Systems: Modeling and Sensitivity Analysis. *Chapter 6 in Stochastic Hybrid Systems (C. G. Cassandras and J. Lygeros, eds.)*, pages 139–167, 2007.

[31] C.G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2008.

[32] C.G. Cassandras, D.L. Pepyne, and Y. Wardi. Optimal control of a class of hybrid systems. *IEEE Trans. on Automatic Control*, 46(3):398 –415, March 2001.

[33] R. Champagnat, P. Esteban, H. Pingaud, and R. Valette. Modeling and Simulation of a Hybrid System through Pr/Tr PN-DAE model. In *Proc. of ADPM'98*, pages 131–137, Reims, France, March 1998.

[34] R. Champagnat, R. Valette, J. C. Hochon, and H. Pingaud. Modeling, simulation and analysis of batch production systems. *Discrete Event Dynamic Systems: Theory and Applications*, 11:119–136, 2001.

[35] C. Chen. *Linear system theory and design*. Oxford University Press, New York, 1984.

[36] H. Chen and A. Mandelbaum. Discrete flow networks: Bottleneck analysis and fluid approximations. *Mathematical Operations Research*, 16:408–446, 1991.

[37] H. Chen and D.D. Yao. *Fundamentals of Queueing Networks. Performance, Asymptotics and Optimization*, volume 46 of *Applications of Mathematics. Stochastic Modelling and Applied Probability*. Springer, 2001.

[38] 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.

[39] F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, 2009.

[40] E.M. Clarke, E.A. Emerson, and J. Sifakis. Model checking: algorithmic verification and debugging. *Communications of the ACM*, 52(11):74–84, 2009.

[41] G. Cohen, S. Gaubert, and J.P. Quadrat. Algebraic system analysis of timed Petri nets. *Idempotency*, page 145, 1998.

[42] 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.

[43] C. Commault, J.M. Dion, and D.H. Trinh. Observability recovering by additional sensor implementation in linear structured systems. In *Procs. of the* 44^{th} *IEEE Conf. on Decission and Control 2005*, pages 7193–7197, Seville, Spain, December 2005.

[44] D. Connors, G. Feigin, and D. Yao. Scheduling semiconductor lines using a fluid network model. *IEEE Trans. on Robotics and Automation*, 10(2):88–98, 1994.

[45] 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.

[46] Y. Dallery, Z. Liu, and D. Towsley. Properties of Fork/Join Queueing Networks with Blocking under various operating mechanisms. *IEEE Trans. on Robotics and Automation*, 13(4):503–518, August 1997.

[47] 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.

[48] R. David and H. Alla. Autonomous and timed Continuous Petri Nets. In *Proc.* 11^{th} *Int. Conf. on Application and Theory of Petri Nets*, pages 367–386, Paris, France, 1990.

[49] R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2004. (Revised 2^{nd} edition, 2010).

[50] I. Demongodin. Generalised Batches Petri Net: Hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1-2):137–162, 2001.

[51] I. Demongodin and A. Giua. Linear programming techniques for analysis and control of Batches Petri nets. In *Procs. of the Int. Workshop on Discrete Event Systems, WODES'10*, pages 4–9, 2010.

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

[53] A. Di Febbraro and A. Giua and G. Menga (eds.). Special Issue on Hybrid Petri Nets. *Discrete Event Dynamic Systems*, 11(1-2), 2001.

[54] J. Ding. *Structural and Fluid Analysis for Large Scale PEPA Models. With Applications to Content Adaptation Systems*. PhD thesis, University of Edinburgh, 2010.

[55] J. Ding and J. Hillston. Fundamental results on fluid approximations of stochastic process algebra models. *Submitted to Performance Evaluation*, 2010.

[56] J.M. Dion, C. Commault, and J. van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, 39(7):1125–1144, 2003.

[57] S. Donatelli, M. Ribaudo, and J. Hillston. A comparison of performance evaluation process algebra and generalized stochastic Petri nets. In *IEEE Petri Nets and Performance Models (PNPM'95)*, pages 158–168, 1995.

[58] D. Down and S.P. Meyn. Piecewise linear test functions for stability and instability of queueing networks. *Queueing Systems*, 27:205–226, 1997.

[59] M. Dub, G. Pipan, and Z. Hanzálek. Stock optimization of a kanban-based assembly line. In *Proc. of the 12th Int. Conf. on Flexible Automation and Intelligent Manufacturing*, 2002.

[60] J. B. Dennis (ed.). *Record of the Project MAC*. Conf. on Concurrent Systems and Parallel Computation. ACM Conf. Records, MIT, June 2-5, 1970.

[61] 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.

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

[63] W. H. Fleming. *Future directions in control theory: a mathematical perspective*. Report of the Panel on Future Directions in Control Theory, SIAM, 1988.

[64] Jay W. Forrester. *Industrial Dynamics*. MIT Press, Cambridge, Mass., 1961.

[65] Jay W. Forrester. *Principles of Systems*. Productivity Press, 1968.

[66] S. Foss and T. Konstantopoulos. An overview of some stochastic stability methods. *Journal of the Operations Research Society of Japan*, 47(4):275–303, 2004.

[67] V. Galpin. Continuous approximation of PEPA models and Petri Nets. *Int. Journal of Computer Aided Engineering and Technology*, 2(4):324–339, 2010.

[68] B. Gaujal and A. Giua. Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica*, 40(9):1505–1516, 2004.

[69] S. Genc and S. Lafortune. Distributed diagnosis of place-bordered Petri nets. *IEEE Trans. on Automation Science and Engineering*, 4(2):206–219, 2007.

[70] D. Gianni and A. D'Ambrogio. A language to enable distributed simulation of extended queueing networks. *Journal of Computers*, 2(4):76–86, June 2007.

[71] A. Giua, F. DiCesare, and M. Silva. Petri Net Supervisors for Generalized Mutual Exclusion Constraints. In 12^{th} *IFAC World Congress*, volume 1, pages 267–270, Sidney, Australia, July 1993.

[72] A. Giua, C. Seatzu, and Y. Wardi. IPA for Continuous Petri Nets. In *Procs. of the 10th Int. Work. on Discrete Event Systems*, pages 57–62. Berlin, Germany, 2010.

[73] L. Habets and JH van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40(1):21–35, 2004.

[74] W. M. Haddad and V. Chellaboina. Stability theory for nonnegative and compartmental dynamical systems with time delay. *Systems and Control Letters*, 51:355–361, 2004.

[75] J. M. Harrison. Stochastic Processing Networks and Activity Analysis. In Y. Suhov, editor, *Analytic Methods in Applied Probability*, In Memory of Fridrik Karpelevich. American Mathematical Society, Providence, RI, 2002.

[76] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian Process Algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, 2010.

[77] R.A. Hayden. Convergence of ODE approximations and bounds on performance models in steady-state. In *9th Workshop on Process Algebra and Stochastically Timed Activities*, 2010.

[78] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. *Formal Methods for Computational Systems Biology*, pages 215–264, 2008.

[79] S. Hennequin, D. Lefebvre, and A. El-Moudni. Fuzzy control of variable speed continuous Petri nets. In *Proc. of the 38th Conf. on Decisions and Control (CDC99)*, 1999.

[80] H. Hermanns, U. Herzog, and J-P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002.

[81] 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.

[82] J. Hillston, L. Recalde, M. Ribaudo, and M. Silva. A Comparison of the Expressiveness of SPA and Bounded SPN models. In *In Procs. of the 9th Int. Workshop on Petri Nets and Performance Models. IEEE Computer Society Press, Aachen*, pages 197–206, September, 2001.

[83] K. Hiraishi. Performance evaluation of workflows using continuous Petri nets with interval firing speeds. *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, 91(11):3219, 2008.

[84] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[85] L.E. Holloway, B.H. Krogh, and A. Giua. A survey of Petri nets methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.

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

[87] M.V. Iordache and P.J. Antsaklis. *Supervisory control of concurrent systems*. Birkhäuser, Boston, 2006.

[88] J. A. Jacquez and C. P. Simon. Qualitative theory of compartmental systems. *SIAM Review*, 35(1):43–79, March 1993.

[89] K. Jensen and L. M. Kristensen. *Coloured Petri Nets. Modelling and Validation of Concurrent Systems*. Springer, Berlin, 2009.

[90] E. Jiménez, J. Júlvez, L. Recalde, and M. Silva. Relaxed Continuous Views of Discrete Event Systems: Petri Nets, Forrester Diagrams and ODES. In *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, The Hague, The Netherlands, October 2004.

[91] E. Jiménez, J. Júlvez, L. Recalde, and M. Silva. On controllability of timed continuous Petri net systems: the join free case. In *Proc. of the 44th IEEE Conf. on Decision and Control (Joint CDC-ECC), Seville, Spain*, 2005.

[92] E. Jiménez, L. Recalde, and M. Silva. 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)*, pages 85–94, 2001.

[93] X. Jing, L. Recalde, and M. Silva. Tracking control of join-free timed continuous Petri net systems under infinite servers semantics. *Journal of Discrete Event Dynamic Systems*, 18(2):263–283, 2008.

[94] X. Jing, L. Recalde, and M. Silva. Tracking control of timed continuous Petri net systems under infinite servers semantics. In *IFAC World Congress*, 2008.

[95] A. Lj. Juloski, W.P.M.H. Heemels, and S. Weiland. Observer design for a class of piecewise linear systems. *Int. Journal of Robust and Nonlinear Control*, 17(15):1387 – 1404, 2007.

[96] J. Júlvez, A. Bemporad, L. Recalde, and M. Silva. Event-Driven Optimal Control of Continuous Petri Nets. In *43rd IEEE Conf. on Decision and Control (CDC)*, pages 69–74, Paradise Island. Bahamas, 2004.

[97] J. Julvez and R.K. Boel. A continuous Petri net approach for model predictive control of traffic systems. *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(4):686–697, 2010.

[98] J. Júlvez, E. Jimenez, L. Recalde, and M. Silva. On Observability and Design of Observers inTimed Continuous Petri Net Systems. *IEEE Trans. on Automation Science and Engineering*, 5(3):532–537, 2008.

[99] J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In W. van der Aalst and E. Best, editors, 24^{th} *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.

[100] 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.

[101] 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.

[102] R. Kara, M. Ahmane, J.J. Loiseau, and S. Djennoune. Constrained regulation of continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 3(4):738–748, 2009.

[103] R. Kara, S. Djennoune, and J.J. Loiseau. State feedback control for the manufacturing systems modeled by continuous Petri nets. In *Proc. of Information Control Problems in Manufacturing (INCOM)*, 2006.

[104] F.P. Kelly, S. Zachary, and I. Ziedins, editors. *Stochastic Networks: Theory and Applications*. Royal Statistical Society LNS, 4. Oxford University Press, 1996 (repr. 2002).

[105] L. Kleinrock. *Queueing Systems. Vol, I.* -. Wiley, New York, 1975.

[106] L. Kleinrock. *Queuing Systems, Volume II: Computer Applications*, volume 2. Wiley, 1976.

[107] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. on Automatic Control*, 53(1):287–297, 2008.

[108] M. Kloetzer, C. Mahulea, C. Belta, and M. Silva. An Automated Framework for Formal Verification of Timed Continuous Petri Nets. *Industrial Informatics, IEEE Trans. on*, 6(3):460–471, 2010.

[109] S. Kumar and P. R. Kumar. Queueing network models in the design and analysis of semiconductor wafer fabs. *IEEE Trans. on Robotics and Automation*, 17(5):548–561, 2001.

[110] T.G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.

[111] D. Lefebvre. Feedback control designs for manufacturing systems modelled by continuous Petri nets. *Int. Journal of Systems Science*, 30(6):591–600, 1999.

[112] D. Lefebvre. Estimation of the firing frequencies in discrete and continuous Petri nets models. *Int. Journal of Systems Science*, 32(11):1321–1332, 2001.

[113] D. Lefebvre, D. Catherine, E. Leclerq, and F. Druaux. Some contributions with Petri nets for the modelling, analysis and control of HDS. *In Proc. of the Int. Conf. on Hybrid Systems and Applications*, 1(4):451–465, 2007.

[114] D. Lefebvre, E. Leclerq, N. El Akchioui, L. Khalij, and E. De Cursi. A geometric approach for the homothetic approximation of stochastic Petri nets. In *Proc. of the 10th IFAC Workshop on Discrete Event Systems*, 2010.

[115] D. Lefebvre, E. Leclerq, L. Khalij, E. De Cursi, and N. El Akchioui. Approximation of MTS stochastic Petri nets steady state by means of continuous Petri nets: A numerical approach. In *Proc. of the 3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, 2009.

[116] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. Gong. Fluid simulation of large scale networks: Issues and tradeoffs. In *Procs. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications*, pages 2136–2142, 1999.

[117] Y. Liu and W. Gong. Perturbation Analysis for Stochastic Fluid Queueing Systems. *Discrete Event Dynamic Systems*, 12(4):391–416, 2002.

[118] D. G. Luenberger. *Introduction to Dynamic Systems. Theory, Models, and Applications*. John Wiley & Sons, 1979.

[119] D.G. Luenberger. An introduction to observers. *IEEE Trans. on Automatic Control*, 16(6):596–602, December 1971.

[120] C. Mahulea, M.P. Cabasino, A. Giua, and C. Seatzu. A state estimation problem for timed continuous Petri nets. In *Decision and Control, 2007 46th IEEE Conf. on*, pages 1770–1775. IEEE, 2008.

[121] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous petri nets. *Automatic Control, IEEE Trans. on*, 53(7):1731–1735, Aug. 2008.

[122] C. Mahulea, A. Ramirez, L. Recalde, and M.Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Trans. on Automation Science and Engineering*, 5(2):307–320, April 2008.

[123] 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.

[124] 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.

[125] C. Mahulea, C. Seatzu, M. P. Cabasino, L. Recalde, and M. Silva. Observer Design for Untimed Continuous Petri Nets. In *American Control Conf.*, pages 4765 – 4770, St. Louis, Missouri, USA, June 2009.

[126] P. Merlin and D. J. Faber. Recoverability of communication protocols. *IEEE Trans. on Communications*, 24(9):1036–1043, 1976.

[127] R. Milner. *A Calculus of Communicating Systems*. Number 92 in Lecture Notes in Computer Science. Springer Verlag, 1980.

[128] V. Misra, W. Gong, and D.F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *In Proc. SIGCOMM*, pages 151–160, 2000.

[129] D. Mitra. Stochastic theory of a fluid model of producers and consumers coupled by a buffer. *Advanced Applied Probability*, 20:646–676, 1988.

[130] M. K. Molloy. Performance Analysis Using Stochastic Petri Nets. *IEEE Trans. on Computers*, 31(9):913–917, 1982.

[131] F.H. Moss and A. Segall. An optimal control approach to Dynamic Routing in Networks. *IEEE Trans. on Automatic Control*, 27(2):329–339, 1982.

[132] T. Murata. Petri Nets: Properties, Analysis and Applications. *Procs. of the IEEE*, 77(4):541–580, 1989.

[133] G. F. Newell. *Applications of Queuing Theory*. Chapman and Hall, 1971 (2nd ed. 1982).

[134] G. Nicolis and I. Prigogine. Fluctuations in nonequilibrium systems. *in Proc. Nat. Acad. Sci. U.S.A.*, 68:2102–2107, 1971.

[135] K. Ogata. *Modern Control Engineering, 4th ed.* Prentice Hall, 2001.

[136] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.

[137] C. A. Petri. *Communication With Automata*. Supplement 1 to Technical Report RADC-TR-65-377, Vol. 1, Griffiss Air Force Base, New York 1966. [Originally published in German: Kommunikation mit Automaten, University of Bonn], 1962.

[138] 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.

[139] L. Recalde and M. Silva. Petri Nets Fluidification revisited: Semantics and Steady state. *APII-JESA*, 35(4):435–449, 2001.

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

[141] 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.

[142] 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.

[143] C. Sauer, E. MacNair, and J. Kurose. The research queueing package: Past, present and future. In *Proc. of the National Computer Conf.*, pages 273–280, 1982.

[144] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[145] C. Seatzu, M.P. Cabasino, C. Mahulea, and M. Silva. New results for fault detection of untimed continuous Petri nets. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conf.. CDC/CCC 2009. Procs. of the 48th IEEE Conf. on*, pages 6952–6957, 2010.

[146] C. Seatzu, C. Mahulea, M.P. Cabasino, and M. Silva. Fault diagnoser design for untimed continuous Petri nets. In *Control Applications (CCA) & Intelligent Control (ISIC), 2009 IEEE*, pages 1598–1604, 2009.

[147] M. Silva. *Las Redes de Petri: en la Automática y la Informática*. Ed. AC, Madrid, 1985.

[148] M. Silva. Towards a Synchrony Theory for P/T Nets. In K. Voss et al., editors, *Concurrency and Nets*, pages 435–460. Springer, 1987.

[149] M. Silva. Introducing Petri Nets. *Practice of Petri Nets in Manufacturing*, pages 1–62, Chapman & Hall, London, 1993.

[150] M. Silva and J. Colom. On the computation of structural synchronic invariants in P/T nets. *Advances in Petri Nets 1988*, pages 386–417, 1988.

[151] 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.

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

[153] 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.

[154] M. Silva and L. Recalde. Unforced continuous Petri nets and positive systems. In Luca Benvenuti, Alberto De Santis, and Lorenzo Farina, editors, *Positive Systems. Procs. of the First Multidisciplinary Int. Symp. on Positive Systems: Theory and Applications (POSTA 2003)*, volume 294 of *LNCIS*, pages 55–62, Rome, August 2003. Springer.

[155] 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.

[156] 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.

[157] 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.

[158] 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.

[159] G. Sun, C. G. Cassandras, and C. G. Panayiotou. Perturbation Analysis of Multiclass Stochastic Fluid Models. *Discrete Event Dynamic Systems*, 14(3):267–307, 2004.

[160] H.J. Sussmann and J. C. Willems. 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, 17(3):32–44, 1997.

[161] E. Teruel and M. Silva. Structure Theory of Equal Conflict Systems. *Theoretical Computer Science*, 153(1-2):271–300, 1996.

[162] K. Trivedi and V. G. Kulkarni. FSPNs: Fluid Stochastic Petri Nets. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *Lecture Notes in Computer Science*, pages 24–31. Springer, Berlin, 1993.

[163] R. Valk, and C. Girault (eds.). *Petri Nets for Systems Engineering. A Guide to Modeling, Verification, and Applications*. Springer, Berlin, 2003.

[164] C.R. Vázquez, A. Ramírez, L. Recalde, and M. Silva. On controllability of timed continuous Petri nets. In *Egerstedt M., Mishra B. eds.: Proc. of the 11th Int. Workshop Hybrid Systems: Computational and Control (HSCC08), LNCS*, volume 4981, pages 528–541, 2008.

[165] C.R. Vázquez, L. Recalde, and M. Silva. Stochastic–continuous state approximation of Markovian Petri net systems. In *Proc. of the 47th IEEE Conf. on Decision and Control (CDC)*, 2008.

[166] C.R. Vázquez and M. Silva. Performance control of Markovian Petri nets via fluid models: A stock-level control example. In *5th IEEE Conf. on Automation Science and Engineering (IEEE CASE)*, 2009.

[167] C.R. Vázquez and M. Silva. Piecewise-linear constrained control for timed continuous Petri nets. In *Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, 2009.

[168] 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, August-September 2010.

[169] C.R. Vázquez, H.Y. Sutarto, R. Boel, and M. Silva. Hybrid Petri net model of a traffic intersection in an urban network. In *IEEE Multiconf. on Systems and Control*, Yokohama, Japan, September 2010.

[170] C.R. Vázquez, J.H. van Schuppen, and M. Silva. A modular-coordinated control for continuous petri nets. Technical report, Universidad de Zaragoza, September, 2010.

[171] M. Vernon, J. Zahorjan, and E. D. Lazowska. A Comparison of Performance Petri Nets and Queueing Network Models. In *3rd Int. Workshop on Modelling Techniques and Performance Evaluation*, Paris, France, March 1987.

[172] G. Walter and M. Contreras. *Compartmental Modeling With Networks*. Birkhauser Boston, 1999.

[173] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time control for structurally persistent continuous petri nets. In *Conf. on Decision and Control CDC 2010*, 2010.

[174] W. Whitt. *Stochastic Process Limits. An introduction to stochastic-process limits and their application to queues*. Series in Operations Research and Financial Engineering. Springer, New York, 2002.

[175] X. Xie. Fluid-stochastic-event graphs for evaluation and optimization of discrete-event systems with failures. *Robotics and Automation, IEEE Transactions on*, 18(3):360–367, 2002.

[176] X. Xu and P.J. Antsaklis. Results and perspectives on computational methods for optimal control of switched systems. In *Hybrid Systems: Computation and Control*, pages 540–555. Springer, 2003.

[177] C. Yao and C. G. Cassandras. Perturbation Analysis and Optimization of Multiclass Multiobjective Stochastic Flow Models. In *IEEE Int. Conf. on Decision and Control, CDC10*, pages 914–919, Shangai, 2010.