Timed Continuous Petri Nets: Quantitative Analysis, Observability and Control

Cristian Mahulea

TESIS DOCTORAL

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Directores:
Manuel Silva Suárez
Laura Recalde Frisón

Septiembre 2007

Timed Continuous Petri Nets: Quantitative Analysis, Observability and Control

Cristian Mahulea

TESIS DOCTORAL

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Directores:
Manuel Silva Suárez
Laura Recalde Frisón

Septiembre 2007

To Liliana

Acknowledgements

I would like to thank all the people who, during the last four years help me in many ways, making possible the present work.

I would like to thank, first and foremost, Manuel Silva and Laura Recalde, my Ph.D. advisors at the University of Zaragoza, Spain. This thesis is based on the work done together in a crucial learning period of my life. I cannot remember how many meetings we had together that clarified me many problems. I am very proud that I had the opportunity to work with two of the most important figures in the field of Petri nets.

Several people have played a decisive role in my work, and I want to mention here Alessandro Giua and Carla Seatzu, professors at the University of Cagliari, that have visited our Department, but also for their hospitality during my stay in their group, in Cagliari. A special thanks to Alessandro that have read part of this thesis at the beginning and gave me tips to improve the presentation.

Thanks a lot to Antonio Ramírez Treviño with which I have worked during his stay in our group. Dear Antonio, I'm looking forward to share a plate of Mexican food with you, not necessarily in a Sunday at the University.

I would like to thank Jorge Júlvez, one of the most intelligent people I have ever met, a mind always wondering, that help me at the beginning of my work to understand better the formalism of continuous Petri nets and made my interest in this particular field to grow.

I cannot forget my supervisor in Romania, Professor Octavian Păstrăvanu, a brilliant teacher and a friend that believed in me and supported me into following this way.

I want to thank the institution that provided me financial support during this thesis: a Ph.D. grant from Spanish Ministry of Science and Education (reference BES-2004-6120), the projects CICYT - FEDER DPI2003-06376 and DPI2006-15390 from the Spanish Ministry of Science and Education and an integrated action Italy-Spain HI2006-0149.

At the personal level, I would like to thank first Manolo and his wife Regina because they help us (me and my wife) to accommodate in Spain. Then, to Juan Pablo, Diego and Jorge, my best friends, for all that they have done for me during my stay in Zaragoza. Thanks to Alessandro and Maria Paola for the moments spent together in Cagliari. Thanks to DIISasters, for all the fun I had in your company, for all the "tapas", for introducing me to "padel" and for allowing me to pretend I was a football player, especially when I was goalkeeper. I cherish all the moments we spent together. I want to thank Javier Campos, for letting me know The Pyrenees in many occasions and Ana Cris for showing me Ezcaray.

I would like to thank all my family in Romania for their support; I want to mention here my grandfather, Ioan. Many thanks to my lovely wife, Liliana that after one year was near me, supporting me and giving me power to finish this thesis. I wish to dedicate this thesis to her.

Redes de Petri fluidas y temporizadas: análisis cuantitativo, observabilidad y control

Resumen

Las redes de Petri constituyen un paradigma muy potente para modelar, analizar y sintetizar sistemas concurrentes basados en eventos. Sin embargo, su análisis y síntesis requiere con frecuencia de algoritmos cuya complejidad crece exponencialmente con el tamaño o marcado de la red. Dada la complejidad computacional subyacente, las relajaciones por fluidificación (redes de Petri Continuas e Híbridas), aunque no siempre utilizables, permiten resolver gran cantidad de casos prácticos. Esta abstracción es útil en el estudio de sistemas procedentes de muchos dominios de aplicación como Sistemas Logísticos, de Fabricación Flexible, de Work Flow Management o redes inalámbricas "ad hoc"; en todos los casos se comparten principios metodológicos para la construcción de modelos, la caracterización de propiedades de buen comportamiento o principios de diseño.

Las redes de Petri continuas, consideradas en esta tesis, difieren de las discretas en que el disparo de las transiciones no está restringido al conjunto de números naturales sino al de los reales positivos. De esta manera, los problemas de programación entera utilizados para estudiar distintas propiedades de las redes discretas se transforman en problemas de programación lineal que tienen una complejidad polinomial. Aunque esto no quiere decir que los problemas son todos polinomiales en complejidad, habiéndolos que son hasta indecidibles.

Este trabajo se alinea en el marco anterior, siendo -en cierto sentido- prolongación de trabajos realizados con carácter previo en las tesis de Laura Recalde y de Jorge Júlvez. Se propone el estudio de cinco temas distintos, todos relacionados con las redes de Petri continuas temporizadas y en general bajo la semántica de infinitos servidores. Los temas propuestos son: (1) semánticas de disparo y monotonía de las prestaciones de la red continua respecto a cambios en la velocidad de disparo de las transiciones y el aumento del marcado inicial (2) transiciones inmediatas para las redes de Petri continuas bajo la semántica de infinitos servidores (3) la observabilidad y la observabilidad óptima para las redes fluidas temporizadas bajo la semántica de infinitos servidores (4) la estimación del estado para las redes fluidas no temporizadas y temporizada bajo la semántica de finitos servidores (5) control en régimen permante y control óptimo del régimen transitorio.

Contents

List of Figures

List of Tables

Chapter 1

Preliminary

Petri Nets (PN) [59] are a discrete event formalism in which the distributed state is a vector of non-negative integers. This is a major advantage with respect to others, such as automata, where the state space is a symbolic unstructured set of global states. This property has been exploited to develop many analysis techniques that do not require the enumeration of the state space (structural analysis) [71]. From a modeling perspective, a key feature of PN is their capacity to graphically represent and visualize primitives such as parallelism, concurrency, synchronization, mutual exclusion, etc.

As other models of concurrent systems, discrete PN may suffer the so called state explosion problem. Therefore the analysis and optimization of these systems require large amount of computational efforts, eventually leading to analytically and computationally untractable problems. One way to tackle this difficulty consists in the relaxation of the original integrality constraints, giving a *fluid* (i.e., continuous) approximation of the discrete event dynamics [28, 68, 69]. Fluid models present one important advantage with respect to discrete ones: the most analysis techniques that are based on integer programming become in the fluid case based on linear programming, and can be solved in polynomial time [7, 42, 62]. Obviously, this does not mean that all problems are polynomial, some of them are undecidible [37]. Fluidification is not a new method; it has been applied also in the case of queuing networks to obtain fluid models [80] used, for example, in computer networks [26].

Fluidification of discrete Petri nets has been introduced independently from three different points of view. In [27] continuization was introduced at *net level*. Some new results, with timing in particular, can be found in [2, 28]. At the same conference in 1987, continuous relaxation was introduced in [67] were the *state equation* was relaxed to find linear programming problems for the efficient computation of non-temporal properties. Further results are presented in [69]. In the case in which the net system is partially fluidified, appear the so called *first order hybrid Petri nets* [5, 6, 7]. In [78], the authors extend the stochastic Petri nets framework to *Fluid Stochastic Petri Nets* by introducing places with continuous tokens and arcs with fluid flow in order to handle stochastic fluid flow systems. No continuous transitions are present in this model. They define hybrid nets in such a way that the discrete and continuous portions may affect each other.

As in the discrete case, continuous Petri nets can be autonomous (untimed) or can have time associated with the transitions or places, namely timed continuous Petri nets (timed contPNs). In the literature, timing is mainly associated to transitions (for place timed continuous Petri nets some works can be found in [32, 21]) and two different ways of defining their firing are often used. Since these two usual definitions are closely related to the semantics used in discrete stochastic PNs, they are called: *finite server semantics* (or *constant speed*) and *infinite server semantics* (or *variable speed*) [28, 68].

This thesis is essentially centred on *timed continuous models*, in particular on *continuous Petri nets under infinite server semantics*. Under this firing semantics, due to the *min* operator occurring in the synchronizations, the continuous model is a multilinear switched dynamic system. Previous results on this class of models were presented in two PhD thesis in Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, and this work is a natural extension of those results.

Reachability aspects and liveness analysis of untimed (autonomous) nets are presented in [61]. The first thesis, devoted only to continuous Petri nets, was [39] where autonomous and timed contPN are studied. Reachability aspects were extended as well as the liveness analysis, but also the steady state performance evaluation is given together with some features on observability and control. The main results of these thesis are recalled in Chapter 2.

The formalism of continuous Petri nets under infinite server semantics is extended including immediate transitions in Section 2.3.2. These transitions appear as a simplification of a timed model when some transitions are *much quicker* than others, case in which they are reduced to *fireable in zero time*. They improve the practical modeling power of the continuous Petri nets under infinite server semantics. The introduction of this kind of transitions means that the simulation of a model has to be done in two steps: after each time step, any enabled immediate transition has to be fired before going on with the timed ones. This problem is discussed in detail in Section 3.5 where a simulation algorithm is provided. Removing immediate transitions at net level becomes a crucial issue in contPNs, because the semantics of timed continuous net models is directly expressible as ODEs with minimum operators, for which reasonable solutions are frequently obtained by means of numerical integration. The basic goal of Section 3.6 is to reduce the number of minimum operators both for any timing (implicitness in the autonomous model) and for a particular timing (implicitness in the timed model).

As was mentioned, for the timed interpretation of continuous Petri nets, two basic definitions of the firing speeds of transitions have been extensively used in the literature: infinite server semantics (or *variable speed*) and finite server semantics (or *constant speed*). Obviously, the modeler would like to know a priori which one will approximate better the discrete model. In previous works we saw that this question depends on the net, and a general answer could not be given [69]. In Section 3.1 we are limiting our study to the class of mono-T-semiflow reducible nets, and prove that one of the definitions is better under very general conditions. This will motivate the choice of infinite server semantics in many parts of the thesis for which more in deep study is performed trying to understand better the system properties.

The first property studied for timed continuous Petri nets under infinite server semantics is *performance monotonicity* with respect to the speed of transitions and with respect to the initial marking. This is usually a desirable property in production systems, for example, if a machine is replaced by a faster one or more resources are available, the production should not decrease. It is shown in Section 3.2 under which conditions these properties hold in the case of mono-T-semiflow reducible nets, a class of nets with a significant modeling power from a practical point of view.

State observation or observability is a very significant question for controller design. For timed continuous Petri nets under infinite server semantics, the state (marking) is supposed to be measurable through sensors, each place being measured with a different one, with different costs. Measuring some places allow the observation of a set of other places ("covered" by that measure). When the set of covered places coincides with the set of all places of the net, we are saying that the contPN system is observable. In [40] observability of contPN was started with a deep study of Join Free nets that can be characterized using linear system theory. Here the observability problem is discussed in the general case and also two new concepts of observability are considered: structural and generic observability. Problems appear for observability with (1) *synchronizations* (or *joins*) that generate switches between linear systems giving non-linear behaviors and (2) *attributions* (or meeting of flows) that for some particular values of the firing rates of the transitions may lead to the loss of observability.

Minimal cost observability of continuous Petri Nets consists in deciding the set of places to be measured such that the net system is observable and the cost function is minimal. This is a covering problem that is NP-complete and a covering algorithm is proposed in chapter 4 using some particular properties of contPN systems that will drastically reduce the complexity.

The results obtained for the observability of systems with infinite server semantics cannot be extended to finite server semantics. Considering that finite server semantics is optimistic [28], i.e., its throughput is in general bigger than the throughput of the discrete net, in chapter 5 we use a particular relaxation of the model assuming that the transitions are not limited to take the biggest admissible value but a value in a given interval and we are trying to determine all states that are consistent with observed sequences of transition firings. The first problem considered here assumes that no observation is available, and thus the set of consistent markings only depends on the time elapsed and we study the observation based on the time-reachability analysis. The problem of state estimation of untimed contPN is studied as well. We show how the results previously obtained for discrete nets can be applied, with minor modifications.

Starting with the crucial question of how to control a contPN system, an approach based on the idea of *slowing down* the firing flow of transitions [69] is considered in Chapters 6 and 7. One question that immediately appears is: given an initial marking, $\boldsymbol{m}_0$, and a constant control action, $\boldsymbol{u}_d$, which steady state $\boldsymbol{m}_d$ is reached? This thesis explores this problem in Chapter 6. For some particular net subclasses, unique solutions are algebraically obtained (thus their characterization is complete). If several steady state markings appear, in many cases they produce the same flow in steady state. In particular, it may be computationally easy (polynomial time) to compute the (maximal) flow vector, even if several (even infinite) steady-state markings may appear.

Computation of an optimal steady state control reference, maximizing a linear profit function that takes into account the throughput, the initial marking and the steady state marking is presented in Section 6.5. If all transitions are controllable, this problem is solved in polynomial time, using a linear programming problem. If some transitions are not controllable the problem becomes more complicated and a Branch & Bound algorithm can be used, like in [42].

The solution proposed here to deal with transitory control is based on a *discrete-time* version of the linear constrained model, thus we need to be sure that the discretization does not produce *spurious* markings, in particular negative markings. To this aim an upper bound on the sampling period is given that guarantees the absence of spurious solutions. Moreover, for the sampled timed contPN, some "equivalence results" regarding the reachability space of sampled timed contPN and (autonomous) contPN are also presented.

Starting from the discrete-time linear model of the contPN we propose an optimal control strategy based on *Model Predictive Control* (MPC) [11], a control method that has become an attractive strategy in the last years. In particular, we investigate the possibility of using both an *implicit* and an *explicit* [10] MPC control strategies. The main advantage of the explicit solution is that the most burdensome part of the procedure is performed off-line. However, as already pointed out in [10], the computational complexity of the explicit approach may become prohibitive when dealing with complex systems which, unfortunately is frequent in the case of contPN models. A comparison among the two procedures is also proposed and the results of various numerical simulations are presented.

Some properties of the system controlled via MPC are discussed also in this thesis, such as feasibility and asymptotic stability. We prove that for contPN systems feasibility is always guaranteed, while asymptotic stability is not ensured. Different approaches are investigated in order to guarantee this property. One of them consists in the introduction of an appropriate terminal constraint, and in such a case asymptotic stability can be guaranteed under appropriate assumptions on the initial state and on the moving horizon.

The contributions of the thesis may be summarized in the following items.

- A semantic for immediate transition in the case of contPN systems under infinite server semantics is introduced and techniques to improve the simulation and analysis of timed contPN are given in Chapters 2 and 3 (a first publication on the topic [62]).

- For the class of mono-T-semiflow reducible nets, performance monotonicity is studied and it is shown under which conditions this property holds. For the same class of nets, it is proved that infinite server semantics provides in general a more accurate approximation of discrete model than finite server semantics. These are illustrated in Chapter 3 (first publications on the topic [53, 54]).

- Observability aspects of timed continuous Petri nets with infinite server semantics and optimal sensor location are shown in Chapter 4 (a first publication on the topic [52]).

- State estimation of untimed continuous Petri nets and timed continuous Petri nets with finite server semantics are considered in Chapter 5 (publication on the topic [15])

- Aspects regarding the controllability and the steady-state control are addressed, together with some characterizations of the equilibrium markings in Chapter 6 (publications on the topic [50, 51]).

- A discrete-time linear constrained model of the controlled timed contPN is derived. It is shown also under which assumptions the sampling period ensures that spurious negative markings do not originate are shown in Chapter 7 (a first publication on the topic [48]).

- The problem of reaching a given steady-state from an initial marking, while minimizing a certain performance index is considered. Two different solutions have been investigated, both based on Model Predictive Control, namely implicit and explicit MPC in Chapter 7 (publications on the topic [33, 34]).

- Feasibility of the Model Predictive Control problem and asymptotic stability of the closed-loop system are analyzed in Chapter 7 (publication on the topic [49]).

The techniques and the algorithms developed through this thesis are implemented under MATLAB and some of them are integrated in the *Continuous Petri Nets Simulator*, a MATLAB based simulator for contPNs developed in our research group. This software can be downloaded from the following address:

`http://webdiis.unizar.es/GISED/gised/contpn/`.

This thesis is structured as follows: in Chapter 2 the basic notions of discrete and continuous Petri nets are given together with some results from previous works and a semantic for immediate transitions. In Chapter 3 performance monotonicity and a comparison between the most used firing semantics for the timed interpretation of contPNs are stated. Also, some techniques to improve the simulation and the analysis of contPN systems under infinite server semantics are given. Chapter 4 deals with some observability aspects of timed contPN with infinite server semantics making a bridge with the observability of piecewise linear systems, also the optimal observability problem is discussed. The state-estimation problem of untimed and timed contPN with finite server semantics is discussed in Chapter 5. Controllability features and steady-state control are illustrated in Chapter 6. In Chapter 7 discrete-time contPN systems are introduced and then used to apply model predictive control. Some conclusions and future directions are drawn in Chapter 8.

Chapter 2

Continuous Petri nets: notations, previous work and first results

Summary

This chapter introduces some basic definitions and concepts related to discrete and continuous Petri nets. Classical results that will be used afterwards are recalled, all of them accompanied by illustrative examples. For the timed interpretation, each firing semantics is discussed in detail and some examples to show their differences are given. In general, the analysis of a general net system is difficult and some of the results hold only for specific subclasses of systems. These subclasses are detailed here, with a more detailed study of mono-T-semiflow reducible nets that give a quite large modeling power. Finally, immediate transitions are introduced in the case of timed continuous Petri net systems under infinite server semantics.

2.1 Discrete Petri nets and the state explosion problem

2.1.1 Basic concepts

Petri nets are a well-known formalism to deal with discrete event systems [29]. Discrete event systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic can be modeled and analyzed by means of Petri nets. As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems.

The reader is assumed to be familiar with Petri nets and for a more deep introduction the following [1, 28, 29, 66] can be consulted. Here, the basic ideas are given.

Definition 2.1. *A Petri net system is a pair* $\langle \mathcal{N}, \boldsymbol{m_0}\rangle$*, where* $\mathcal{N}$ *is a net,* $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ *is a PN.* P *and* T *are disjoint (finite) sets of places and transitions, and* $\boldsymbol{Pre}$ *and* $\boldsymbol{Post}$ *are* $|P| \times |T|$ *sized, natural valued, incidence matrices, and* $\boldsymbol{m}_0$ *is the initial marking.*

A PN can be graphically represented as a weighted bipartite graph. The nodes are represented by the set of places drawn as circles and by the set of transitions drawn as rectangles. These nodes are connected with a set of oriented arcs, each arc can connect a place with a transition or a transition with a place. $\boldsymbol{Pre}[p,t] = w > 0$ means that there is an arc from p to t with *weight* (or *multiplicity*) w, and $\boldsymbol{Post}[p,t] = w > 0$ means that there is an arc from t to p with *weight* w. The classical concepts of graph theory, as connectedness, strong connectedness can be directly applied to PN. For a node $v \in P \cup T$, the sets of its input and output nodes are denoted as ${}^\bullet v$, and $v^\bullet$, respectively. Each place can contain a natural number of tokens, this number represents the *marking* of the place. The initial distribution of tokens is called *initial marking* and is denoted, in general, by $\boldsymbol{m}_0$.

A transition t is *enabled* at $\boldsymbol{m}$ iff for every $p \in {}^\bullet t, \boldsymbol{m}[p] \geq \boldsymbol{Pre}[p,t]$. Its *enabling degree* measures the maximal firing of the transition that can be done in one step,

$$enab(t,\boldsymbol{m}) = \min_{p\in{}^\bullet t}\left\{\left\lfloor \frac{m[p]}{\boldsymbol{Pre}[p,t]}\right\rfloor\right\}.$$

The firing of t in a certain amount $\alpha \in \mathbb{N}$, $\alpha \leq \text{enab}(t,\boldsymbol{m})$ leads to a new marking $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[P,t]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow matrix* and we say that $\boldsymbol{m}'$ is reachable from $\boldsymbol{m}$. This firing is also denoted by $\boldsymbol{m}[t(\alpha)\rangle\boldsymbol{m}'$ or by $\boldsymbol{m}_0 \xrightarrow{\alpha t} \boldsymbol{m_1}$.

If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a sequence σ, a *state* (or *fundamental*) *equation* can be written:

$$\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma}, \tag{2.1}$$

where $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}$ is the firing count vector. The set of all markings reachable from $\boldsymbol{m}_0$, or reachability set is denoted by $RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$.

If the number of tokens in a place is always less than or equal to a natural number, then the place is *bounded*. A PN system is *bounded* when every place is bounded ($\forall p \in P, \exists b_p \in \mathbb{N}$ with $\boldsymbol{m}[p] \leq b_p$ at every reachable marking $\boldsymbol{m}$). A net $\mathcal{N}$ is *structurally bounded* when $\langle \mathcal{N}, \boldsymbol{m_0}\rangle$ is bounded for every initial marking $\boldsymbol{m}_0$

2.1.2 Structural concepts

Annulers of the incidence matrix are important because they induce certain invariant relations which are useful for reasoning about the behavior. *Flows* (*semiflows*) are integer (natural) annulers of $\boldsymbol{C}$. Right and left annulers are called T- and P-(semi)flows, respectively. We call a semiflow $\boldsymbol{v}$ minimal when its support, $||\boldsymbol{v}||$, i.e., the set of its non-zero components, is not a proper superset of the support of any other semiflow, and the g.c.d. of its elements is one.

If $\boldsymbol{y} \geq \boldsymbol{0}$ is such that $\boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}$ then, every marking $\boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ satisfies: $\boldsymbol{y} \cdot \boldsymbol{m} = \boldsymbol{y} \cdot \boldsymbol{m}_0$. This provides a "token balance law". Analogously, if $\boldsymbol{x} \geq \boldsymbol{0}$ is such that $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$, then $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{m}_0$. That is, T-semiflows correspond to *potential* repetitive sequences that brings the system from $\boldsymbol{m}_0$ back to $\boldsymbol{m}_0$.

If $\boldsymbol{x} > \boldsymbol{0}$ exists such that $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$, the net is said to be *consistent*, and if $\boldsymbol{y} > \boldsymbol{0}$ exists such that $\boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}$, the net is said to be *conservative*.

Traps and siphons are structural dual concepts with high importance in the analysis of many net properties as deadlock-freeness. A set of places, Θ, is a *trap* iff $\Theta^\bullet \subseteq {}^\bullet\Theta$. In discrete net systems traps when marked cannot get emptied. Analogously, a set of places, Φ, is a *siphon* iff ${}^\bullet\Phi \subseteq \Phi^\bullet$. One interesting property is that empty siphons will unavoidably remain empty throughout all the evolution of the net system.

2.1.3 Liveness and deadlock-freeness

A transition t is *live* iff it can ultimately occur from every reachable marking, i.e., for every $\boldsymbol{m} \in RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$, $\boldsymbol{m}' \in RS^{ut}(\mathcal{N}, \boldsymbol{m})$ exists such that t is enabled in $\boldsymbol{m}'$. A PN system $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is *live* if every transition is live. Liveness ensures that no single action in the system can become unattainable.

A PN is *deadlock-free* when any reachable marking enables some transition. Clearly, deadlock-freeness is a necessary condition for liveness.

A PN is *structurally live* (*deadlock-free*) if an initial marking $\boldsymbol{m}_0$ exists for which the net system $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is live (deadlock-free).

2.1.4 Petri net subclasses

Two transitions $t, t' \in T$ are in *conflict relation* at marking $\boldsymbol{m}$ iff there exist $k, k' \in \mathbb{R}_{>0}$ such that $\boldsymbol{m} \geq k \cdot \boldsymbol{Pre}[P,t]$ and $\boldsymbol{m} \geq k' \cdot \boldsymbol{Pre}[P,t']$, but $\boldsymbol{m} \not\geq k \cdot \boldsymbol{Pre}[P,t] + k' \cdot \boldsymbol{Pre}[P,t']$. For this, it is necessary that ${}^\bullet t \bigcap {}^\bullet t' \neq \emptyset$ and in that case it is said that t and t' are in structural conflict relation. The structural conflict relation is not transitive, and the *c*oupled conflict relation is defined as its transitive closure. Each equivalence class is called a coupled conflict set, that will be denoted as CCS_i, and $SCCS$ will represent the set of all the coupled conflict sets. A particular kind of conflicts are those in which the preconditions of the transitions are all the same. We will say that t_i and t_j are in *continuous equal conflict relation*, if a constant k exists such that for all $p \in P$, $\boldsymbol{Pre}[p,t_i] = k \cdot \boldsymbol{Pre}[p,t_j]$. This is an equivalence relation on the set of transitions and each equivalence class in an equal conflict set denoted, for a given t, $EQS(t)$. $SEQS$ is the set of all the equal conflict sets of a given net.

The nets can be classified according to their structure:

- $\mathcal{N}$ is *ordinary* if all the arc weights are unitary, i.e., $\forall p_i \in P$, $\forall t_j \in T$, $\boldsymbol{Pre}[p_i,t_j] \leq 1$ and $\boldsymbol{Post}[p_i,t_j] \leq 1$;

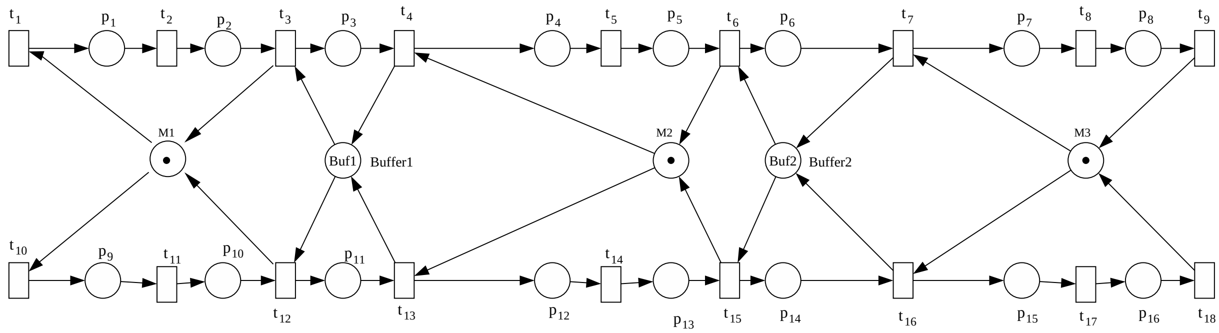


Figure 2.1: Continuous (mono-T-semiflow reducible) net system.

- $\mathcal{N}$ is a *Marked Graph* (MG) if it is ordinary and $\forall p \in P : |p^\bullet| = |^\bullet p| = 1$.
- $\mathcal{N}$ is a *Join Free* (JF) if $\forall t \in T$, $|^\bullet t| \leq 1$;
- $\mathcal{N}$ is a *weighted T-graph* if $\forall p \in P : |p^\bullet| = |^\bullet p| = 1$;
- $\mathcal{N}$ is a *Choice Free* (CF) net if $\forall p \in P$, $|p^\bullet| \leq 1$;
- $\mathcal{N}$ is a *Free Choice* (FC) net if it is ordinary and $\forall t, t' \in T$, if $^\bullet t \cap {}^\bullet t' \neq \emptyset$ then $^\bullet t = {}^\bullet t'$;
- $\mathcal{N}$ is an *Equal Conflict* (EQ) net iff $\forall t_1, t_2 \in T$ such that $^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$, $\boldsymbol{Pre}[P, t_1] = \boldsymbol{Pre}[P, t_2]$.
- $\mathcal{N}$ is an *Continuous Equal Conflict* (CEQ) net iff $\forall t_1, t_2 \in T$ such that $^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$, $\boldsymbol{Pre}[P, t_1] = k \cdot \boldsymbol{Pre}[P, t_2]$.

The class that will be studied in more detail in this thesis is called mono-T-semiflow reducible. The definition of this class is based on the concept of mono-T-semiflow nets.

Definition 2.2. *$\mathcal{N}$ is a mono-T-semiflow net if it is consistent, conservative, and has only one minimal T-semiflow, $\boldsymbol{x}$ (i.e. given a transition t_i there is only one T-semiflow such that $\boldsymbol{x}[t_i] = 1$).*

2.1.5 State explosion and fluidification

The state explosion appears frequently in discrete event systems, making enumerative analysis methods practically inapplicable. This problem affects also discrete Petri nets.

Example 2.3. *Let us consider the PN system in Fig. 2.1. The size of the reachability sets for different initial markings of $Buffer_1$ and $Buffer_2$ is given in Table 2.1. Clearly, the size is growing, for example if $Buf_1 = Buf_2 = 10$ we have more than 500.000 reachable states. If the values are further increased, the analysis techniques based on the enumeration will be quite difficult to apply.*

Fluidification is an approximation technique that relaxes the description by removing the integrality constraints. Applying this idea to the discrete PNs, the firing of the transitions

Table 2.1: The size of the reachability set of the net in Fig. 2.1.

Buf_1, Buf_2	Number of reachable markings
1	1125
2	4500
3	12500
4	28125
5	55125
10	544500

is not limited to natural numbers but to positive real numbers leading to continuous Petri nets [28, 68].

In a continuous Petri net the firing of a transition is not constrained to be a natural number, but a nonnegative real number. Thus, when a transition is fired, a real amount of tokens is removed from the input places and some amount is put in the output places. This way, the marking of a continuous Petri net becomes a vector of nonnegative real numbers, where the dimension of the vector is equal to the number of places. (In the discrete case, the marking was a vector of natural numbers). Therefore, in continuous Petri nets, the transitions can be seen as valves through which "fluid tokens" flow, and the places can be seen as deposits in which this fluid is stored. Exhaustive enumeration techniques have no sense in continuous Petri nets, since the set of reachable markings is not a discrete set anymore, but a continuous region.

Continuization of Petri nets offers not only the advantage of avoiding the state explosion problem, but also gives the chance of using linear programming techniques (that can be solved in polynomial time) instead of integer programming problems (which entail an exponential complexity).

2.2 Untimed Continuous Petri nets

2.2.1 Definition

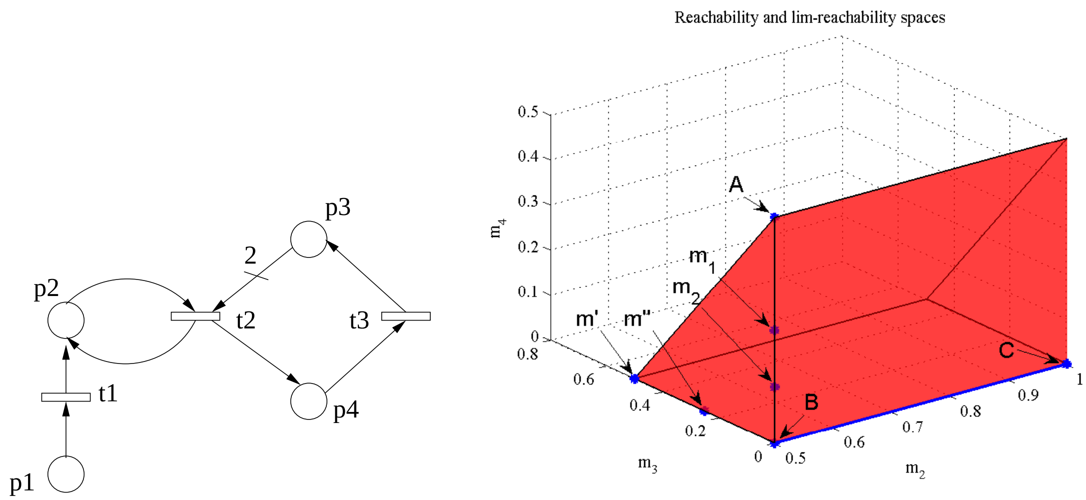
In this section we will introduce the concept of continuous Petri net.

Definition 2.4. *A contPN system is a pair $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$, where $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is a net structure (defined as in discrete case: with a set of places P, a set of transitions T, and the pre and post incidence matrices $\boldsymbol{Pre}$ and $\boldsymbol{Post}$), and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial marking (distributed state).*

A transition $t_j \in T$ is enabled at $\boldsymbol{m}$ iff $\forall p_i \in {}^\bullet t_j$, $m_i > 0$. Since in the continuous case, the marking of a place is not limited to a natural number, the enabling degree is not limited to a natural number as in discrete case and will be given by:

$$enab(t, \boldsymbol{m}) = \min_{p \in {}^\bullet t} \left\{ \frac{m[p]}{\boldsymbol{Pre}[p,t]} \right\}.$$

An enabled transition t_j can fire in any amount $0 < \alpha < enab(t_j, \boldsymbol{m})$ leading to a new marking $\boldsymbol{m}' = \boldsymbol{m} + \alpha \boldsymbol{C}[P,t]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token-flow matrix.*



(a) Untimed ContPN used in Example 2.7

(b) Reachability and lim-reachability sets

Figure 2.2: Illustration of reachability and lim-reachability sets.

The structural concepts given in Section 2.1.2 for discrete nets keep the same definition in the case of continuous nets.

2.2.2 Reachability

The set of markings that are reachable with a finite firing sequence for a given system $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is denoted as $RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$. It is defined as:

Definition 2.5. *[64] $RS^{ut}(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} |$ a finite fireable sequence $\sigma = \alpha_1 t_{a1} \cdots \alpha_k t_{ak}$ exists such that $\boldsymbol{m}_0 \xrightarrow{\alpha_1 t_{a1}} \boldsymbol{m_1} \xrightarrow{\alpha_2 t_{a2}} \boldsymbol{m_2} \cdots \xrightarrow{\alpha_k t_{ak}} \boldsymbol{m_k} = \boldsymbol{m}\}$ where $t_{ai} \in T$ and $\alpha_i \in \mathbb{R}^+_{\geq 0}$.*

An interesting property of $RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$ is that it is a *convex set* [64]. That is, if two markings $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ are reachable, then any marking $\boldsymbol{m_3} = \alpha \cdot \boldsymbol{m_1} + (1-\alpha) \cdot \boldsymbol{m_2}$, $\forall \alpha \in [0,1]$ is also reachable. Therefore, the reachability space of a continuous PN is at least the convex hull of the discrete one.

Reachability may be extended to *lim-reachability* assuming that infinitely long sequences can be fired. From the point of view of the analysis of the behavior of the system, it is interesting to consider these markings since in the limit the system may converge to them.

Definition 2.6. *[64] Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a continuous system. A marking $\boldsymbol{m} \in \mathbb{R}^+_{\geq 0}$ is* lim-reachable *iff 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-reachability set is the set of lim-reachable markings, and will be denoted $lim - RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$.

Example 2.7. *Let us consider the contPN in Fig. 2.2a with $\boldsymbol{m_0} = [0.5, 0.5, 0, 0.5]^T$. At this marking, either t_1 or t_3 can be fired. The firing of t_3 in the amount 0.5 leads to the new marking $\boldsymbol{m}' =$*

$[0.5, 0.5, 0.5, 0]^T$ *from which* t_2 *can be fired in an amount* 0.25 *leading to* $\boldsymbol{m_1} = \left[0.5, 0.5, 0, \frac{0.5}{2}\right]^T$. *Now firing* t_3 *in an amount of* 0.25 *followed by* t_2 *in the maximum possible amount* 0.125 *the system evolves to* $\boldsymbol{m_2} = \left[0.5, 0.5, 0, \frac{0.5}{4}\right]^T$. *It is obvious that keeping firing the same pair of transitions* t_3 *and* t_2*, at step* k *the marking obtained is:* $\boldsymbol{m_k} = \left[0.5, 0.5, 0, \frac{0.5}{2^k}\right]^T$*, for which* $\lim_{k\to\infty} \boldsymbol{m_k} = [0.5, 0.5, 0, 0]^T = \boldsymbol{m}$. *Therefore, the marking* $\boldsymbol{m}$ *is reached in the untimed contPN system at the limit.*

In Fig. 2.2b are shown the reachability set and the lim-reachability set for the contPN in Fig. 2.2a. The only difference between these two sets is the segment $[B, C]$ *that belongs to the* $lim - RS^{ut}$ *but not to* RS^{ut}*. It can be seen that going from* A *to* B *firing* t_3 *and* t_2*, at every step the distance is halved but the point* B *is never reached (like in Zeno paradox).*

Characterizations of RS^{ut} and $lim - RS^{ut}$ are given in [41]. To give the characterizations, let us denote as $FS(\mathcal{N}, \boldsymbol{m}_0)$ the set of all sets of transitions for which there exists a sequence fireable from $\boldsymbol{m}_0$ that contains those and only those transitions.

Definition 2.8. *[41]* $FS(\mathcal{N}, \boldsymbol{m}_0) = \{v|$ *there exists a sequence fireable from* $\boldsymbol{m}_0$*,* σ*, such that* $v = ||\boldsymbol{\sigma}||\}$.

Then, a full characterization of the reachable markings is obtained.

Theorem 2.9. *[41] A marking* $\boldsymbol{m} \in RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$ *iff*

1. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{\sigma} \geq \boldsymbol{0}$
2. $||\boldsymbol{\sigma}|| \in FS(\mathcal{N}, \boldsymbol{m}_0)$
3. *there is no empty trap in* $\mathcal{N}_{\boldsymbol{\sigma}}$ *at* $\boldsymbol{m}$

where $\mathcal{N}_{\boldsymbol{\sigma}}$ *denotes the subnet obtained from* $\mathcal{N}$ *removing the transitions not in the support of* $\boldsymbol{\sigma}$ *and the isolated places that appear.*

For the $lim - RS^{ut}$, the characterization is quite similar, only the third condition disappears. This is because in continuous systems, marked trap can be emptied in the limit.

Theorem 2.10. *[41] A marking* $\boldsymbol{m} \in lim - RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$ *iff*

1. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{\sigma} \geq \boldsymbol{0}$
2. $||\boldsymbol{\sigma}|| \in FS(\mathcal{N}, \boldsymbol{m}_0)$.

Example 2.11. *Going back to the Example 2.7 we have seen that* $\boldsymbol{m} = [0.5, 0.5, 0, 0]^T$ *is a lim-reachable marking from* $\boldsymbol{m}_0 = [0.5, 0.5, 0, 0.5]^T$. *The first two conditions of Theorem 2.9 and the conditions of Theorem 2.10 are satisfied because taking* $\boldsymbol{\sigma} = [0, 0.5, 1]^T$, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$ *and* $||\boldsymbol{\sigma}|| \in FS(\mathcal{N}, \boldsymbol{m}_0)$. *This is a lim-reachable marking, because the third condition of Theorem 2.9 is not satisfied: a marked trap is empty at* $\boldsymbol{m}$ *(*$\Theta = \{p_3, p_4\}$, $\Theta^\bullet = {}^\bullet\Theta = \{t_2, t_3\}$*).*

In many cases, the systems under discussion have some interesting properties as consistency. In this case, the lim-reachability condition can be relaxed to a linear inequality system.

Proposition 2.12. *[41, 64] Let* $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ *be a contPN system. If it is consistent and* all transitions are fireable *the following statements are equivalent:*

1. $\boldsymbol{m}$ *is lim-reachable*
2. $\exists \boldsymbol{\sigma} \geq 0$ *s.t.* $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \geq \boldsymbol{0}$
3. $\boldsymbol{B}_y^T \cdot \boldsymbol{m} = \boldsymbol{B}_y^T \cdot \boldsymbol{m}_0$, $\boldsymbol{m} \geq 0$ *where* $\boldsymbol{B}_y$ *is a basis of P-flows.*

2.2.3 Implicit arcs and places

In discrete models, a place is said to be implicit if it can be removed without changing the behavior of the system [71]. For continuous models it is important even to know if a place is never the unique that bounds the enabling degree. We will see that for timed contPN the simulation and the analysis techniques are easier if the number of joins (transitions with more than one input place) is reduced. Therefore, for the output arcs of a place the notion of implicit arc is defined. If all the output arcs of a place are implicit, this place will be implicit, and can be completely removed. In that case we will say that the place is implicit.

Definition 2.13. *An arc* (p, t) *is implicit in* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *if* p *is never the unique place that defines the enabling of* t*. That is, for every reachable marking* $\boldsymbol{m}$, $\frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p,t]} \geq \frac{\boldsymbol{m}[p']}{\boldsymbol{Pre}[p',t]}$ $\forall p' \in {}^{\bullet}t$.
Place p *is implicit if all its output arcs are implicit.*

This is equivalent to saying that for any reachable marking $\boldsymbol{m}$, the following system has no solution, where s represents the amount in which t is fired.

$$\left\{ \begin{array}{l} \boldsymbol{m}[P'] - \boldsymbol{Pre}[P', t] \cdot s \geq \boldsymbol{0},\ P' = P \setminus \{p\} \\ \boldsymbol{m}[p] - \boldsymbol{Pre}[p, t] \cdot s < 0 \\ s \geq 0 \end{array} \right. \tag{2.2}$$

Using that any reachable marking has to verify the state equation, a sufficient condition for (p, t) being implicit is that the solution to the following linear programming problem (LPP), z, verifies $\boldsymbol{m}_0[p] \geq z$.

$$\begin{array}{rl} z = \max & \boldsymbol{Pre}[p, t] \cdot s - \boldsymbol{C}[p, T] \cdot \boldsymbol{\sigma} \\ s.t. & \boldsymbol{m} - \boldsymbol{C} \cdot \boldsymbol{\sigma} = \boldsymbol{m}_0 \\ & \boldsymbol{m}[P'] - \boldsymbol{Pre}[P', t] \cdot s \geq \boldsymbol{0},\ P' = P \setminus \{p\} \\ & \boldsymbol{m}, \boldsymbol{\sigma}, z \geq 0 \end{array} \tag{2.3}$$

Notice that $\boldsymbol{m} = \boldsymbol{m}_0, \boldsymbol{\sigma} = \boldsymbol{0}$ and $s = 0$ is always solution of the equations in (2.3). Hence, applying duality, the solution to this LPP is the same as the solution of its dual problem:

$$\begin{array}{rl} z = \min & \boldsymbol{y}^T \cdot \boldsymbol{m}_0 \\ s.t. & \boldsymbol{y}^T \cdot \boldsymbol{C}[P', T] \leq \boldsymbol{C}[p, T] \\ & \boldsymbol{y}^T \cdot \boldsymbol{Pre}[P', t] \geq \boldsymbol{Pre}[p, t] \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array} \tag{2.4}$$

For place p to be implicit, putting together all the equations related to its output arcs, a sufficient condition is that $\boldsymbol{m}_0[p] \geq z$, with z defined as follows:

$$\begin{array}{rl} z = \min & \boldsymbol{y}^T \cdot \boldsymbol{m}_0 \\ s.t. & \boldsymbol{y}^T \cdot \boldsymbol{C}[P', T] \leq \boldsymbol{C}[p, T] \\ & \boldsymbol{y}^T \cdot \boldsymbol{Pre}[P', p^{\bullet}] \geq \boldsymbol{Pre}[p, p^{\bullet}] \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array} \tag{2.5}$$

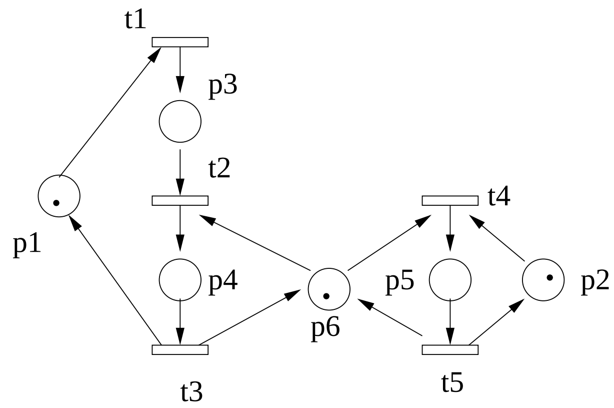


Figure 2.3: ContPN system used in Example 2.14.

This condition (equivalent to that for discrete net systems in [66]) simplifies the one in [71], because traps can be emptied.

Example 2.14. *Let us consider the contPN system in Fig 2.3. For this net and initial marking,* p_2 *is implicit because it is easy to observe that* $m_2(\tau) = m_4(\tau) + m_6(\tau)$. *Hence* p_2 *will never constrain the firing of* t_4 *and can be removed without changing the system evolution.*

2.2.4 Configurations and regions

Definition 2.15. *A* configuration *of* $\mathcal{N}$ *is a set of* (p, t) *arcs, one per transition covering the set of transitions* T.

Remark 2.16. *Abusing notation, a configuration will also represent the set of places associated to the arcs in the configuration. We use this denomination if there is no confusion.*

The number of configurations of $\mathcal{N}$ is given by the net structure:

$$\gamma = \prod_{t \in T} |{}^{\bullet}t| \tag{2.6}$$

The reachability space of a contPN can be partitioned, associating to each configuration $\mathcal{C}_k$ a *region* R_k, i.e., $(lim-)RS^{ut} = R_1 \cup \cdots \cup R_\gamma$. Some regions can be eventually empty in the case in which some places are implicit. A region R_k is defined as the set of markings such that if (p_i, t_j) is an arc belonging to the configuration $\mathcal{C}_k$ then,

$$\forall \boldsymbol{m} \in R_k, \frac{\boldsymbol{m}[p_i]}{\boldsymbol{Pre}[p_i, t_j]} = \min_{p_l \in {}^{\bullet}t_j} \frac{\boldsymbol{m}[p_l]}{\boldsymbol{Pre}[p_l, t_j]}.$$

These regions only may have the borders in common, but their insides are disjoint. In fact, the regions represent a partition (except on the border) of the reachability space.

Example 2.17. *The net in Fig. 2.3 has* $\gamma = 4$ *configurations:*

- $\mathcal{C}_1 = \{(p_1, t_1), (p_3, t_2), (p_4, t_3), (p_2, t_4), (p_5, t_5)\}$
- $\mathcal{C}_2 = \{(p_1, t_1), (p_6, t_2), (p_4, t_3), (p_2, t_4), (p_5, t_5)\}$

- $\mathscr{C}_3 = \{(p_1, t_1), (p_3, t_2), (p_4, t_3), (p_6, t_4), (p_5, t_5)\}$

- $\mathscr{C}_4 = \{(p_1, t_1), (p_6, t_2), (p_4, t_3), (p_6, t_4), (p_5, t_5)\}$

According to the Example 2.14, with this initial marking, p_2 is implicit, and so $\boldsymbol{m}[p_2] \geq \boldsymbol{m}[p_6]$, therefore the regions associated to $\mathscr{C}_1$ and $\mathscr{C}_2$ are empty, except at the border when $m_2 = m_6$.

2.2.5 Liveness and deadlock-freeness

Using the similitude with the discrete nets, the liveness and the deadlock-freeness definitions are given immediately:

Definition 2.18. *[41] Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a contPN system.*

- $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ lim-deadlocks *iff a marking $\boldsymbol{m} \in \text{lim}-RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$ exists such that $enab(t, \boldsymbol{m}) = 0$ for every transition t.*

- $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ *is* lim-live *iff for every transition t and for any marking $\boldsymbol{m} \in \text{lim} -RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$ a successor $\boldsymbol{m}'$ exists such that $enab(t, \boldsymbol{m}') > 0$.*

- $\mathcal{N}$ *is* structurally lim-live *iff $\exists \boldsymbol{m}_0$ such that $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is* lim*-live.*

Although $deadlocks$ at limit may only be reached in the limit, they represent an important system weakness. They enable the system to reach a marking in which all transitions have infinitely small enabling degrees.

In general, if a system is live and bounded as discrete, it is not necessary live and bounded as continuous, a kind of reverse can be proved under weak conditions.

Theorem 2.19. *[64] Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a bounded lim-live P/T system. Then, $\mathcal{N}$ is structurally live and structurally bounded as a dicrete net.*

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 as continuous. In particular, the *rank theorem* is a necessary condition based on the existence of left and right annullers of the token flow matrix and the existence of an upper bound on the rank of this matrix.

Theorem 2.20. *[71] A bounded and lim-live contPN is consistent, conservative and $rank(\boldsymbol{C}) \leq |SEQS| - 1$.*

For some subclasses of continuous nets, there are other necessary conditions. In the case of mono-T-semiflow contPN systems, lim-liveness is equivalent with deadlock-freeness.

Theorem 2.21. *[43] A continuous mono-T-semiflow system is lim-live iff it is lim-deadlock-free.*

Using this result, and Theorem 2.20, the necessary condition can be obtained:

Theorem 2.22. *[43] Let $\mathcal{N}$ be a mono-T-semiflow net. If $\mathcal{N}$ is structurally lim-live then every transition has at least one input persistent place, i.e., that has only one output transition.*

2.3 (Unforced) Timed Continuous Petri nets

2.3.1 Finite and infinite server semantics

If a timed interpretation is included in the model, the fundamental equation depends on time: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$. Differentiating with respect to time, the following equation is obtained: $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing sequence will be called the *(firing) flow* of the timed model: $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$.

To associate a time semantics to a transition, observe that a transition can be seen as a station in Queuing Networks (QNs): the meeting point of servers and clients. In discrete nets a transition can model a station with one server (*single server semantics*), k servers working in parallel (*multiple server semantics*) or an infinite number of servers (*infinite server semantics*). Single and multiple (both *finite*) server semantics for t_j can be simulated with infinite server by adding a self-loop place p_j around t_j with the appropriate number of tokens. Thus infinite server semantics is more general for discrete models.

Different definitions of the flow of continuous timed transitions have been given, the two most important being *finite server* (or *constant speed*) and *infinite server* (or *variable speed*) [2, 68]. Under *finite server* (constant speed), each transition t_j has associated a real positive number, λ_j, called *maximal firing speed*. If the markings of the input places of the transition are strictly greater than zero (*strongly enabled*), its flow will be constant, equal with this value (all servers working at full speed). Otherwise (*weakly enabled*), the flow will be the minimum between its *maximal firing speed* and the total input flow to the places with zero marking. With this definition, λ_j represents the product of the number of servers in the transition and their speed.

$$f_j = \begin{cases} \lambda_j, \text{ if } \nexists p_i \in {}^\bullet t \text{ with } m_i = 0 \\ \min\left\{ \min\limits_{p_i \in {}^\bullet t | m_i = 0} \left\{ \sum\limits_{t' \in {}^\bullet p_i} \frac{\boldsymbol{f}[t'] \cdot \boldsymbol{Post}[t', p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}, \lambda_j \right\} \text{otherwise} \end{cases} \tag{2.7}$$

For the computation of the instantaneous firing speed of the transitions under finite server semantics different procedures have been proposed. In [28], an iterative algorithm is given, while in [7] a linear programming problem (LPP) is used. In fact, in [7], the continuous transitions have associated two values: a minimum and a maximum firing speed. When the transitions are strongly enabled, their instantaneous flow can be any value in the given interval. This can be seen as a relaxation of finite server semantics and we will discuss more of this formalism in chapter 5. However, taking the lower bound equal to zero, keeping the conflict policy and maximizing the instantaneous flow, the obtained semantics is equivalent with finite server semantics. The only difference is in the case of empty cycles but, since this constructions cannot model a system with good properties, we are using here the LPP approach. The instantaneous firing speed $\boldsymbol{f}$ of the transitions at marking $\boldsymbol{m}$ in the case in which there is no choice can be computed using:

$$\begin{array}{lll} \max & \{\mathbf{1}^T \cdot \boldsymbol{f}\} & \\ \text{s.t.} & 0 \le f_i \le \lambda_i & \forall t_i \in T \\ & \boldsymbol{C}[p, T] \cdot \boldsymbol{f} \ge \mathbf{0} & \forall p \text{ with } \boldsymbol{m}[p] = 0 \end{array} \tag{2.8}$$

This LPP computes the maximum instantaneous firing speed where there is no conflict, so maximizing the sum of flows with the constraints corresponding to (2.7). Observe that

(2.7) is not defining completely the flow of a contPN under finite server semantics (it is not deterministic). In the case of conflict, a resolution policy should be specified. Otherwise, (2.8) can have many solutions. When using priorities for the routing policy, the computation of the firing speed implies solving several LPP [7]. In each step the previous computed flows are fixed (added as new constraints) and a new problem maximizing the flow of the transitions with greatest priority not computed before is solved. If t_1, t_2 are in CEQ relation it is also easy to define how the flow is split at the conflict, just adding a constraint of the form $\alpha \cdot f_1 = (1-\alpha) \cdot f_2$ in (2.8). If t_1 and t_2 are not in CEQ relation the splitting is more difficult to introduce.

Under *infinite server* (variable speed) the flow of a transition t_j can be expressed as:

$$f_j = \lambda_j \cdot enab(t_j, \boldsymbol{m}) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{\boldsymbol{Pre}[p_i, t_j]} \right\} \tag{2.9}$$

The enabling degree of the transition t_j represents the number of active servers for that transition at $\boldsymbol{m}$. The flow will be the number of active servers times the work each one does per time unit, i.e., λ_j. Notice that the number of active servers depends only on the weighted marking of the input places.

In both cases, the timed system is defined by the net and a positive vector, $\boldsymbol{\lambda}$ (here it is assumed that there are no immediate transitions), although $\boldsymbol{\lambda}$ has a different meaning under each semantics: it is the firing rate of a transition in the case of infinite server semantics, and it is a maximal firing speed in the case of finite server semantics (the product of the number of servers and the firing rate of one server).

Example 2.23. *Let us consider the PN system in Fig. 2.3, modeling a shared resource (place p_6) among two processes. Observe that in this case the behavior of the discrete PN is the same for finite and infinite server semantics because the servers are implicit in the model. The speed of each transition is defined as $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]$ (for both firing semantics).*

*Let us consider first the model under the continuous relaxation, and **infinite server semantics**. The flows through transitions given by* (2.9) *are:*

$$\begin{cases} f_1(\tau) = m_1(\tau) \\ f_2(\tau) = 2 \cdot \min\{m_3(\tau), m_6(\tau)\} \\ f_3(\tau) = m_4(\tau) \\ f_4(\tau) = \min\{m_2(\tau), m_6(\tau)\} \\ f_5(\tau) = 0.5 \cdot m_5(\tau) \end{cases} \tag{2.10}$$

*Assume now **finite server semantics**. At $\boldsymbol{m}_0$, the input places of t_1 and t_4 are marked, therefore t_1 and t_4 are strongly enabled and $f_1 = f_4 = 1$. The other transitions are weakly enabled and their flow depends on the input flows to the empty input places. For t_2, the input flow to p_3 (the only empty input place) is* 1*, hence $f_2 = \min\{\lambda_2, 1\} = 1$. Transition t_3 will also work at its maximum speed because the input flow in p_4 is $f_2 = 1$, equal to λ_1. For t_5, the input flow to p_5 is* 1*, then its flow is limited by its maximal firing speed that is* 0.5*. Therefore:*

$$\begin{cases} f_1(\tau) = f_2(\tau) = f_3(\tau) = f_4(\tau) = 1 \\ f_5(\tau) = 0.5 \end{cases} \tag{2.11}$$

The same result is obtained solving the LPP defined in (2.8)*:*

$$\begin{array}{rl} & \max\{\mathbf{1}^T \cdot \boldsymbol{f}\} \\ s.t. & f_1 \le 1 \\ & f_2 \le 2 \\ & f_3 \le 1 \\ & f_4 \le 1 \\ & f_5 \le 0.5 \\ & f_2 \le f_1 \\ & f_3 \le f_2 \\ & f_5 \le f_4 \end{array} \tag{2.12}$$

These values are kept constant until $\tau = 2$ *when* m_6 *becomes empty and a new computation of the flows is required to ensure the positiveness of the markings.*

Piecewise linear behaviors are obtained under finite and infinite server semantics, i.e., technically speaking, both continuous timed models are *hybrid*. Under finite server semantics the behavior changes when a place is emptied, and flows are piecewise constant functions. Under infinite server semantics the change of behavior happens when in a synchronization the place that gives the enabling degree changes. In this case the flow is a differential piecewise linear function of the marking. In both situations, the switching among the linear systems is given by internal events.

Under infinite server semantics, the behavior of a system without synchronizations would be linear with constraints. In general, the set of linear systems is defined by the pairs transition-input place that defines its flow. Using the notions of *configuration* and *region* defined in Subsection 2.2.4, to each region a linear system is associated that will govern the evolution of the contPN when the marking belongs to that region. This can be done since in each region, the same set of places gives the enabling degree of the transitions.

For a given region R_k, we can define the *constraint matrix* $\mathbf{\Pi_k} : T \times P \to \mathbb{R}_{\ge 0}$ such that:

$$\mathbf{\Pi_k}[t_j, p_i] = \begin{cases} \dfrac{1}{\boldsymbol{Pre}[p_i, t_j]}, & \text{if } (\forall \boldsymbol{m} \in R_k),\ \dfrac{m_i}{\boldsymbol{Pre}[p_i, t_j]} = \min\limits_{p_h \in {}^\bullet t_j} \left\{ \dfrac{m_h}{\boldsymbol{Pre}[p_h, t_j]} \right\}; \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

Example 2.24. *For the system sketched in Fig. 2.4 with* $\boldsymbol{\lambda} = \mathbf{1}$*, the flow of* t_1 *can be restricted by the marking of* p_1 *or* p_4 *and the flow of* t_2 *can be restricted by the marking of* p_2 *or* p_4*. Thus, the number of regions in this case is* $\gamma = 4$ *and they are defined as follows:*

- R_1: $\dfrac{m_1}{2} \le \dfrac{m_4}{2}$ *and* $m_2 \le m_4$ *with* $\mathbf{\Pi}_1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$,
- R_2: $\dfrac{m_1}{2} \le \dfrac{m_4}{2}$ *and* $m_2 \ge m_4$ *with* $\mathbf{\Pi}_2 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$,
- R_3: $\dfrac{m_1}{2} \ge \dfrac{m_4}{2}$ *and* $m_2 \le m_4$ *with* $\mathbf{\Pi}_3 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$,

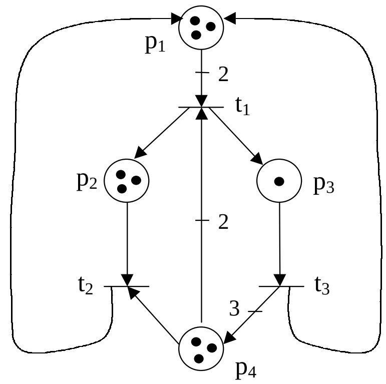


Figure 2.4: ContPN system.

- R_4: $\frac{m_1}{2} \geq \frac{m_4}{2}$ *and* $m_2 \geq m_4$ *with* $\mathbf{\Pi}_4 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

■

If marking $\boldsymbol{m}$ belongs to R_k, we denote $\mathbf{\Pi}(\boldsymbol{m}) = \mathbf{\Pi}_k$ the corresponding constraint matrix. Furthermore, the firing rate of transitions can also be represented by a diagonal matrix $\mathbf{\Lambda}$: $T \times T \to \mathbb{R}_{>0}$, where

$$\mathbf{\Lambda}[t_j, t_h] = \begin{cases} \lambda_j \text{ if } j = h \\ 0, \text{ otherwise} \end{cases}$$

Using this notation, the non-linear flow of the transitions at a given marking $\boldsymbol{m}$ (see eq. (2.9) for f_j) can be written as:

$$\boldsymbol{f} = \mathbf{\Lambda} \cdot \mathbf{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} \tag{2.14}$$

A contPN system evolves and may reach a steady state (i.e. a marking such that $\dot{\boldsymbol{m}}(\tau) = 0$). The configuration (see Definition 2.15 and Remark 2.16) of the steady state marking will be called the *steady state configuration*. Given an initial marking, the problem of whether the system approaches a steady state is undecidible [37]. However this steady state marking, if it exists, has to fulfill certain conditions: the flow it defines has to be a T-semiflow (because $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{0}$), and has to verify the equations defined by the P-flows (necessary for reachability). That is,

$$\begin{cases} \boldsymbol{B}^T \cdot \boldsymbol{m}_0 = \boldsymbol{B}^T \cdot \boldsymbol{m} \\ \boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{0} \\ f_j = \lambda_j \cdot \min\limits_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{\boldsymbol{Pre}[p_i, t_j]} \right\}, \forall t_j \in T \\ \boldsymbol{f} \geq \boldsymbol{0} \end{cases} \tag{2.15}$$

where $\boldsymbol{B}$ is a basis of P-flows.

In other words, the solutions of these equations represent all the possible ways of distributing the tokens of the P-flows so that the system remains on that marking. However, it may happen that several markings fulfill these conditions. We will refer to all these possible steady states markings as *possible equilibrium markings*. All their configurations will be called *possible equilibrium configurations*.

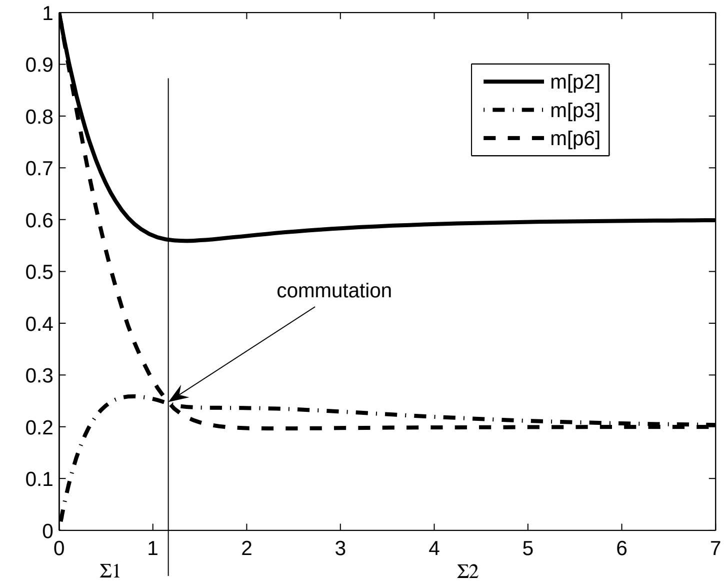


Figure 2.5: Evolution of contPN in Fig. 2.3 with $\boldsymbol{\lambda} = [1,2,1,1,0.5]^T$ under infinite server semantics.

Example 2.25. *Let us go back to Example 2.23 (system of Fig. 2.3) under the same conditions:* $\boldsymbol{\lambda} = [1,2,1,1,0.5]$*. It is evidenced in Example 2.14 that* p_2 *is implicit and may be removed.*

Infinite server semantics*. Because* p_2 *is implicit, in* (2.10)*,* $f_4 = \min\{m_2, m_6\} = \min\{m_4 + m_6, m_6\} = m_6$*, and so two configurations can govern the system evolution. At* $\tau = 0$*,* $m_3 < m_6$*, therefore the evolution of the contPN system is governed by the configuration* $\mathscr{C}_3 = \{(p_1,t_1), (p_3,t_2), (p_4,t_3), (p_6,t_4), (p_5,t_5)\}$ *(see Example 2.17). It leads to the following linear system:*

$$\Sigma_1 : \begin{cases} \dot{m}_1(\tau) &=& f_1(\tau) - f_2(\tau) = m_1(\tau) - 2m_3(\tau) \\ \dot{m}_2(\tau) &=& f_5(\tau) - f_4(\tau) = 0.5m_5(\tau) - m_6(\tau) \\ \dot{m}_3(\tau) &=& f_1(\tau) - f_2(\tau) = m_1(\tau) - 2m_3(\tau) \\ \dot{m}_4(\tau) &=& f_2(\tau) - f_3(\tau) = 2m_3(\tau) - m_4(\tau) \\ \dot{m}_5(\tau) &=& f_4(\tau) - f_5(\tau) = m_6(\tau) - 0.5m_5(\tau) \\ \dot{m}_6(\tau) &=& f_3(\tau) + f_5(\tau) - f_2(\tau) - f_4(\tau) \\ &=& m_4(\tau) + 0.5m_5(\tau) - 2m_3(\tau) - m_6(\tau) \end{cases}$$

The evolution of the contPN system is sketched in Figure 2.5. It evolves according to Σ_1 *until* $\tau \simeq 1.14$ *t.u. when* $m_3(\tau) = m_6(\tau)$*. At that point, a switch occurs and the new governing configuration is:* $\mathscr{C}_4 = \{(p_1,t_1),(p_6,t_2),(p_4,t_3),(p_6,t_4),(p_5,t_5)\}$*. It leads to a new linear system:*

$$\Sigma_2 : \begin{cases} \dot{m}_1(\tau) &=& m_1(\tau) - 2m_6(\tau) \\ \dot{m}_2(\tau) &=& 0.5m_5(\tau) - m_6(\tau) \\ \dot{m}_3(\tau) &=& m_1(\tau) - 2m_6(\tau) \\ \dot{m}_4(\tau) &=& 2m_6(\tau) - m_4(\tau) \\ \dot{m}_5(\tau) &=& m_6(\tau) - 0.5m_5(\tau) \\ \dot{m}_6(\tau) &=& m_4(\tau) + 0.5m_5(\tau) - 2m_6(\tau) - m_6(\tau) \end{cases}$$

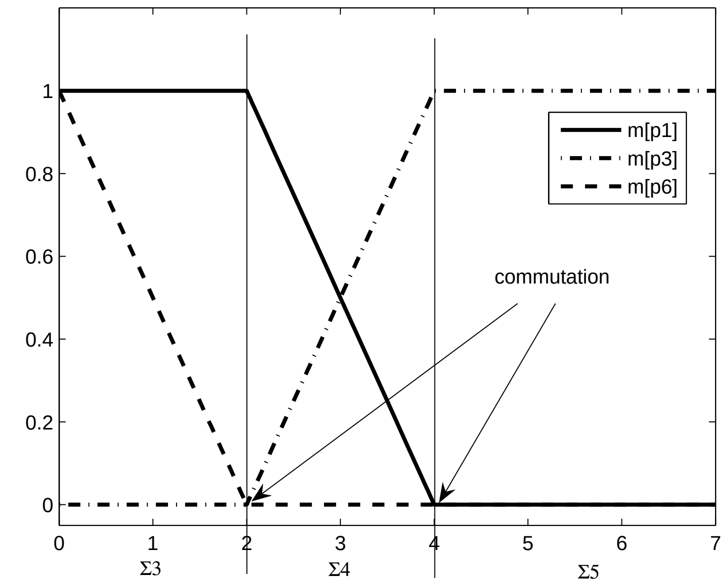


Figure 2.6: Evolution of contPN in Fig. 2.3 with $\boldsymbol{\lambda} = [1,2,1,1,0.5]^T$ under finite server semantics.

The system evolves according to Σ_2 *and reaches the steady state marking* $[0.4, 0.6, 0.2, 0.4, 0.4, 0.2]^T$ *with the corresponding flow:* $[0.4, 0.4, 0.4, 0.2, 0.2]^T$.

Finite server semantics*. The evolution of the system under finite server semantics is presented in Fig. 2.6. As explained before, at* $\boldsymbol{m}_0$*, the equations of the system will be:*

$$\Sigma_3 : \begin{cases} \dot{m}_1(\tau) = f_1(\tau) - f_2(\tau) = 0 \\ \dot{m}_2(\tau) = f_5(\tau) - f_4(\tau) = -0.5 \\ \dot{m}_3(\tau) = f_1(\tau) - f_2(\tau) = 0 \\ \dot{m}_4(\tau) = f_2(\tau) - f_3(\tau) = 0 \\ \dot{m}_5(\tau) = f_4(\tau) - f_5(\tau) = 0.5 \\ \dot{m}_6(\tau) = f_3(\tau) + f_5(\tau) - f_2(\tau) - f_4(\tau) = -0.5 \end{cases}$$

These equations hold until $\tau = 2$*, when* m_6 *and* m_2 *become empty. At this time, the marking is* $[1,0,0,0,1,0]$*. Now,* t_1 *and* t_5 *are strongly enabled, therefore* $f_1 = 1$ *and* $f_5 = 0.5$*. The weakly enabled transitions* t_2 *and* t_4 *are in conflict and a resolution policy should be specified. Assume, for example, that the flow of* t_2 *is equal to the flow of* t_4*. Moreover, the output flows of all the empty places are upper bounded by the input flows. This leads to the following system of equations:*

$$
\max\{\mathbf{1}^T \cdot \boldsymbol{f}\}
\begin{cases}
f_1 = 1 & \textit{(strongly enabled)}\\
f_5 = 0.5 & \textit{(strongly enabled)}\\
f_2 = f_4 & \textit{(conflict resolution)}\\
f_4 \le f_5 & \textit{(}p_2 \textit{ is empty)}\\
f_2 \le f_1 & \textit{(}p_3 \textit{ is empty)}\\
f_3 \le f_2 & \textit{(}p_4 \textit{ is empty)}\\
f_2 + f_4 \le f_3 + f_5 & \textit{(}p_6 \textit{ is empty)}\\
f_3 \le 1 & \textit{(maximal firing speed)}\\
f_2 \le 2 & \textit{(maximal firing speed)}\\
f_4 \le 1 & \textit{(maximal firing speed)}
\end{cases}
\tag{2.16}
$$

Remark that the inequality obtained for p_2 empty,i.e., $f_4 \le f_5$ is redundant and can be obtained from the inequalities for p_4 and p_6. This happens because place p_2 is implicit in model. The solution of (2.16) is $f_2 = f_3 = f_4 = 0.5$. So, the system of equations that defines the evolution after $\tau = 2$ is:

$$
\Sigma_4 : \begin{cases}
\dot{m}_1(\tau) = -0.5\\
\dot{m}_2(\tau) = \dot{m}_4(\tau) = \dot{m}_5(\tau) = \dot{m}_6(\tau) = 0\\
\dot{m}_3(\tau) = 0.5
\end{cases}
$$

At $\tau = 4$, p_4 is emptied and a new flow computation has to be done. The current marking is $[0,0,1,0,1,0]$. The only strongly enabled transition is t_5, hence $f_5 = 0.5$. Writing a linear programming problem as done before, $f_1 = f_2 = f_3 = f_4 = 0.5$ is obtained. These values correspond to a steady state marking ($\dot{\boldsymbol{m}}(\tau) = 0$).

Clearly, the evolution of a contPN system is quite different under both semantics: different throughputs and different steady state markings are obtained.

Other semantics may appear in a natural way in particular application domains. For example, in population dynamic models the product of the markings of the input places of a transition may be useful to express the probabilities that tuples of elements meet in a synchronization [68]. This semantics can be viewed as the result of decoloring some colored nets under infinite server semantics.

Notice also that not every transition can be reasonably fluidified. For example, in a traffic system, the “power on” or “power off” of a semaphore is purely discrete and in many cases can be inappropriate to continuize it. If some transitions remain discrete and some are continuous then the model is conceptually *hybrid* [28, 69].

2.3.2 Immediate transitions

Immediate transitions appear as a simplification of a timed model when some transitions are “much quicker” than others, case in which they are reduced to “fireable in zero time”. In a certain sense, transitions could be logically classified, according to their speeds, into timed (producing a job in a sensible time) and immediate (leading to some vanishing markings). Of course, immediate transitions may be defined at several levels of immediateness, but here we will simplify the presentation assuming that transitions are either timed or immediate, i.e., $T = T_T \cup T_I$, $T_T \cap T_I = \emptyset$.

While immediate transitions were considered in contPN under finite server semantics [28], they have not been defined in the case of infinite server semantics. Under finite server semantics, it is ensured that the immediate transitions are weakly enabled at the initial marking (each one has at least one empty input transition) and its flow depends on the input flow in the empty input places. Using (2.7), if t_j is an immediate transition, its flow is defined as:

$$f_j = \min_{p_i \in {}^\bullet t | m_i = 0} \left\{ \sum_{t' \in {}^\bullet p_i} \frac{\boldsymbol{f}[t'] \cdot \boldsymbol{Post}[t', p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}$$

For infinite server semantics, the problem is different since in (2.9) the flow of a transition is by definition the product of the enabling degree times its firing rate, that it is infinite now.

In discrete markovian models immediate transitions can be removed when the Markov Chain is generated. It is a kind of compilation technique because the tangible Markov chain (in which no vanishing state remains) will not suffer, in general, from stiffness anymore. Reduction of immediate transitions at net level for discrete Petri nets also gain some attention in the literature [1]. The advantage obtained in the case in which the immediate transitions are removed derives from the isomorphism between the reachability graph of the PN model without immediate transitions, and the state transition rate diagram of the underlying markovian model. In the context of timed continuous net systems, obtaining stiffness-free models is particularly interesting if numerical integration has to be done, because the net model and the derived ODE system are isomorphic. Even if the introduction of immediate transitions was historically done in order to cope with stiffness, since the *routing* of clients in service networks is much quicker than the treatments to be performed, they can be seen at a logical modelling level as a way to decouple routing from services. A real positive number is associated to each transition $t \in T_I$ that is in conflict. This number is represented as $\boldsymbol{r}[t]$ and defines the firing rates of the transitions in a conflict. That is, if t_1 and t_2 are both enabled and in conflict relation, the firing of t_1 divided by the firing of t_2 will be equal to $\boldsymbol{r}[t_1]/\boldsymbol{r}[t_2]$.

In Section 3.5 it is presented an algorithm that permits to simulate contPN systems under infinite server semantics that contain immediate transitions while in Section 3.6 some techniques to reduce the number of immediate transitions are presented.

2.3.3 Timed implicit arcs

In subsection 2.2.3 we have seen that it is better if the implicit places, i.e., places that have all the output arcs implicit, are removed from the model. This will help when the simulation and the analysis of the underlying model is performed. There, only the autonomous model has been considered, not taking time into account. Hence the results will be valid for any timing of the model. However, timing will introduce additional restrictions in the dynamics of the system, and so arcs that were not implicit in general might be so for a particular timing. A new concept, of *timed implicit arcs* is introduced.

It is not easy to give general rules to characterize this kind of implicit arcs. However, some situations can be dealt with. Assume for example a net with a par-begin par-end in which the synchronization is an immediate transition (see Fig. 2.7). It is clear that in the empty par-begin par-end net ($t_{start} - t_{syn}$), one of the arcs in the synchronization will be implicit. In this case, since $\boldsymbol{\lambda}[T_{par1}] = 2$ and $\boldsymbol{\lambda}[T_{par2}] = 1$, the arc (p_5, t_{syn}) is implicit. Even more, in this case place p_5 becomes timed implicit, and can be removed. Its marking can always be deduced using that $\boldsymbol{m}[p_5] = \boldsymbol{m}[p_4] + \boldsymbol{m}[p_6] - \boldsymbol{m}[p_3]$.

This can be generalized to the case in which some tokens appear in one place of the par-begin par-end subnet. To simplify, let us consider first a net with only a par-begin par-end

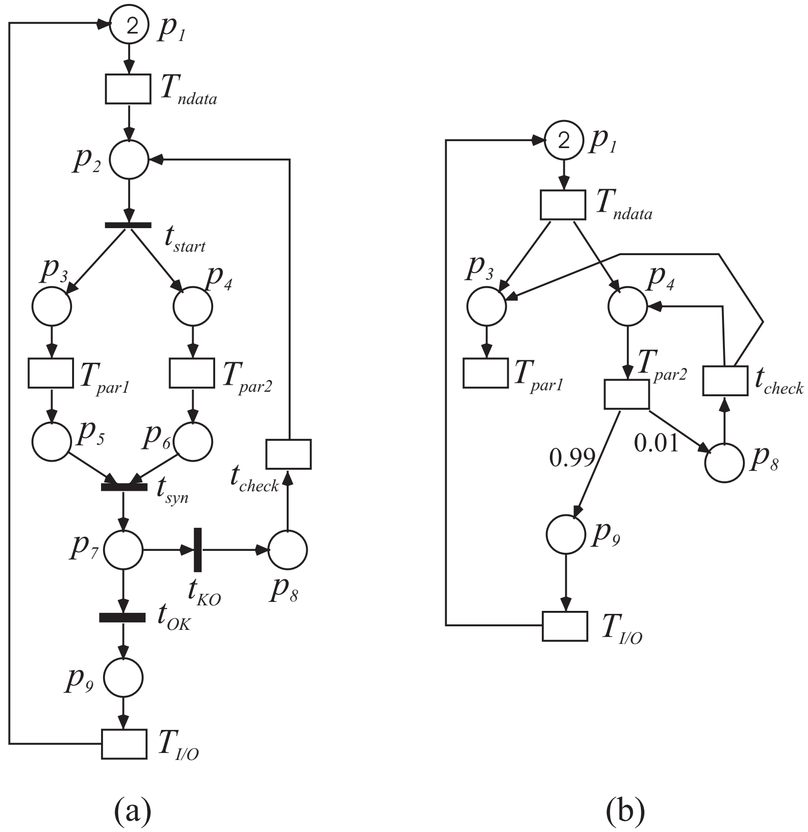


Figure 2.7: (a) Net system (taken from [1]) and (b) its reduction as a contPN.

connected by a place (see Fig. 2.8). If $\boldsymbol{\lambda}[t_2] \leq \boldsymbol{\lambda}[t_3]$, then for any $q \geq 0$, the arc (p_5, t_4) will be timed implicit. If $\boldsymbol{\lambda}[t_2] > \boldsymbol{\lambda}[t_3]$, it may also happen that (p_5, t_4) is timed implicit if q is large enough. To get the lower q for this, we need to ensure that always $\boldsymbol{m}[p_4] = 0$ and $\boldsymbol{m}[p_5] \geq 0$. At the beginning, the evolution of the system can be described by the following equations:

$$\begin{aligned}
&\dot{\boldsymbol{m}}[p_1] = -\boldsymbol{\lambda}[t_1] \cdot \boldsymbol{m}[p_1] + \boldsymbol{\lambda}[t_2] \cdot \boldsymbol{m}[p_2] \\
&\dot{\boldsymbol{m}}[p_3] = \boldsymbol{\lambda}[t_1] \cdot \boldsymbol{m}[p_1] - \boldsymbol{\lambda}[t_3] \cdot \boldsymbol{m}[p_3] \\
&\boldsymbol{m}[p_1] + \boldsymbol{m}[p_2] = k \\
&\boldsymbol{m}[p_1] + \boldsymbol{m}[p_3] + \boldsymbol{m}[p_5] = k + q
\end{aligned}$$

Integrating the system, it can be deduced that if $q \geq k \cdot \frac{1/\boldsymbol{\lambda}[t_2] - 1/\boldsymbol{\lambda}[t_3]}{1/\boldsymbol{\lambda}[t_2] + 1/\boldsymbol{\lambda}[t_1]}$, then $\boldsymbol{m}[p_5] \geq 0$.

Of course, in general the net will not be so simple. However, it can be proved that, as long as the flow of transition t_1 in the original net is at most that of t_1 in the basic par-begin par-end net, if (p_5, t_4) is implicit in the basic net it will also be implicit in the complete net. Hence, the idea is to find $\boldsymbol{\lambda}[t_1]$ and k such that the relationship between the flows of t_1 in both nets is guaranteed. For example, a simple way is to define k as the maximum number of tokens in an input place of t_1 (of course, if there are more than one place, take the minimum among them) and $\boldsymbol{\lambda}[t_1]$ the same as it is in the original net or, if this transition were immediate, a previous timed one.

In this schema it has been assumed that the synchronization was immediate. In fact, this is no restriction since the case with a timed synchronization can be reduced to this one

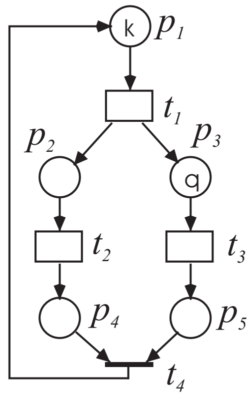


Figure 2.8: Par-begin par-end net

by unfolding the timed transition into an immediate synchronization followed by the timed transition.

2.3.4 Steady state performance Bounds

For computing an upper bound of the flow of a transition in steady state the following non-linear programming problem presented in [42] can be used:

$$\begin{aligned}
&\max\{\phi[t] \mid \boldsymbol{\mu}_{ss} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma}, && (a)\\
&\quad \phi_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\min_{p\in{}^\bullet t}\left\{\frac{\boldsymbol{\mu}_{ss}[p]}{\boldsymbol{Pre}[p,t]}\right\}, \forall t\in T, && (b)\\
&\quad \boldsymbol{C}\cdot\phi_{ss} = \boldsymbol{0}, && (c)\\
&\quad \boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \boldsymbol{0}\}. && (d)
\end{aligned} \tag{2.17}$$

The equations in (2.17) represent: (a) the state equation that should be satisfied by the marking; (b) the flow definition for each transition; (c) the steady state condition; (d) the firing vector and the flow should be positive.

Nevertheless, this non-linear programming problem is difficult to solve due to the "min" condition. When a transition t has a single input place, the equation 2.17 (b) reduces to (2.18). When t has more than an input place, it can be relaxed (linearized) as in (2.19).

$$\phi_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\boldsymbol{Pre}[p,t]}, \text{if } p = {}^\bullet t \tag{2.18}$$

$$\phi_{ss}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\boldsymbol{Pre}[p,t]}, \forall p\in{}^\bullet t, \text{otherwise} \tag{2.19}$$

This way we obtain a single linear programming problem, that can be solved in polynomial time:

$$\begin{aligned}
&\max\{\phi[t] \mid \boldsymbol{\mu}_s s = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma},\\
&\quad \phi_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\boldsymbol{Pre}[p,t]}, \text{if } p = {}^\bullet t,\\
&\quad \phi_{ss}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\boldsymbol{Pre}[p,t]}, \forall p\in{}^\bullet t, \text{otherwise},\\
&\quad \boldsymbol{C}\cdot\phi_{ss} = \boldsymbol{0},\\
&\quad \boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \boldsymbol{0}\}.
\end{aligned} \tag{2.20}$$

Unfortunately, this LPP is a relaxation of the original problem (2.17), and provides in general a non-tight bound, i.e. the solution may be non-reachable for any distribution of the tokens verifying the P-semiflow load conditions, $\boldsymbol{y} \cdot \boldsymbol{m}_0$. This can happen for a join transition t since a solution of (2.20) can be obtained such that its flow, $\phi_{ss}[t]$, is strictly less than its enabling degree multiplied by $\boldsymbol{\lambda}[t]$, so the real flow of t is greater and the steady state condition, i.e., $\boldsymbol{C} \cdot \phi_{ss}$, may not be satisfied.

One way to improve this bound is to force the equality for at least one place per synchronization. This corresponds to a correct interpretation of the *min* operator. The problem is that there is no way to know in advance which of the input places should restrict the flow. A branch & bound algorithm can be used to compute a steady state marking that fulfills what (2.17) expresses. The procedure is: first solve (2.20). If the marking solution of (2.20) does not correspond to a steady state (i.e. there is at least one transition (z in total) such that all its input places have “more than necessary” tokens) choose one of those synchronizations and solve the set of LPPs that appear when each one of the input places are assumed to be defining the flow. Variable eqs tracks the information of the places assumed to define the flow of some synchronizations. That is, build a set of LPPs by adding an equation that relates the marking of each input place with the flow of the transition. These subproblems become children of the root search node. The algorithm is applied recursively, generating a tree of subproblems. If the optimal marking to a subproblem is a feasible steady state marking, it can be used to prune the rest of the tree: if the solution of the LPP for a node is smaller than the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned.

Algorithm 2.26. ***procedure*** BRANCH & BOUND*($\mathcal{N}$,$\boldsymbol{m}_0$,eqs)*

```
    Global variable bound := 0
    (x, z) := max_LPP(N, m0, eqs)
    if x ≤ bound or the LPP was unfeasible then                ▷ Do nothing.
    else
        if z = 0 then                      ▷ The solution represents a steady state
            bound := max(bound, x)
        else
            take a t ∈ nt do
            for every p ∈ •t do
                eqs := eqs ∪ (p, t)
                Branch & Bound(N, m0, eqs)
            end for
        end if
    end if
end procedure
```

2.3.5 Mono-T-semiflow reducible systems

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 t_j is visited (fired) for each visit to (firing of) the reference transition t_i. In general, the visit ratio of a

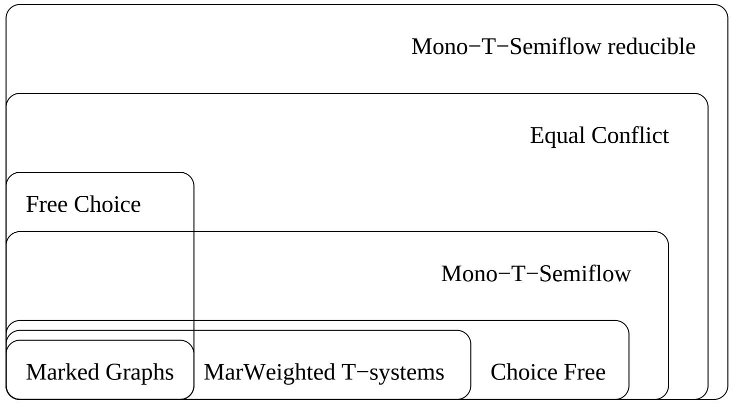


Figure 2.9: Live and bounded net classes included in the class of mono-T-Semiflow reducible nets.

discrete Petri net depends on the structure, the rates of the transitions, and the initial marking [17].

Mono-T-semiflow nets is the class of nets for which the visit ratio vector depends only on the net structure (the token flow matrix). The class of *mono-T-semiflow reducible* nets is the class for which the visit ratio vector depends at most on the net and the firing rate vector, but not on $\boldsymbol{m}_0$, i.e., $\boldsymbol{v}^{(i)} = \boldsymbol{v}^{(i)}(\mathcal{N}, \boldsymbol{\lambda})$.

Definition 2.27. *[42] $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is a* mono-T-semiflow reducible *net if it is consistent, conservative and the following system has a unique solution:*

$$\begin{cases} \boldsymbol{C} \cdot \boldsymbol{v}^{(1)} = \boldsymbol{0} \\ \frac{\boldsymbol{v}^{(1)}[t_i]}{\boldsymbol{Pre}[p,t_i] \cdot \lambda[t_i]} = \frac{\boldsymbol{v}^{(1)}[t_j]}{\boldsymbol{Pre}[p,t_j] \cdot \lambda[t_j]} & \forall t_i, t_j \text{ in CEQ relation}, \forall p \in {}^{\bullet}t_i \\ \boldsymbol{v}^{(1)}[t_1] = 1 \end{cases}$$

Notice that checking if a net belongs to the class has a polynomial time complexity. Moreover, it offers a reasonable modelling power, since it includes the class of live and bounded EQ nets [76] (which are the weighted generalization of FC nets), live and bounded CF nets [75], live and bounded weighted T-graph nets and live and bounded MG nets. The inclusions have been represented in Fig. 2.9. Additionally, synchronized processes with shared resources can be modeled with mono-T-semiflow reducible net models and deterministically synchronized sequential processes (DSSP) [63].

Example 2.28. *The mono-T-semiflow reducible net in Fig. 2.10 represents a queuing network, adapted from [18] and explained in detailed in Ex. 2.10. It has four minimal T-semiflows:* $\boldsymbol{x_1} = t_1 + t_2 + t_3$, $\boldsymbol{x_2} = t_1 + t_2 + t_4 + t_6 + t_8 + t_{11}$, $\boldsymbol{x_3} = t_6 + t_8 + t_{10}$ *and* $\boldsymbol{x_4} = t_1 + t_2 + t_5 + t_7 + t_9 + t_{12}$. *The values of* λ_3, λ_4 *and* λ_5 *will determine the splitting of the flow entering in* p_3 *(because* t_3, t_4 *and* t_5 *are in free-choice relationship) while* λ_{10} *and* λ_{11} *will define the splitting of the flow entering in* p_{11}. *For example, for the particular value of* $\boldsymbol{\lambda} = \boldsymbol{1}$, *the visit ratio vector normalized for* t_3 *is* $\boldsymbol{v}^{(3)} = [3,3,1,1,1,2,1,2,1,1,1,1]^T$ *(the addition of the minimal T-semiflows).*

An immediate consequence of Definition 2.27:

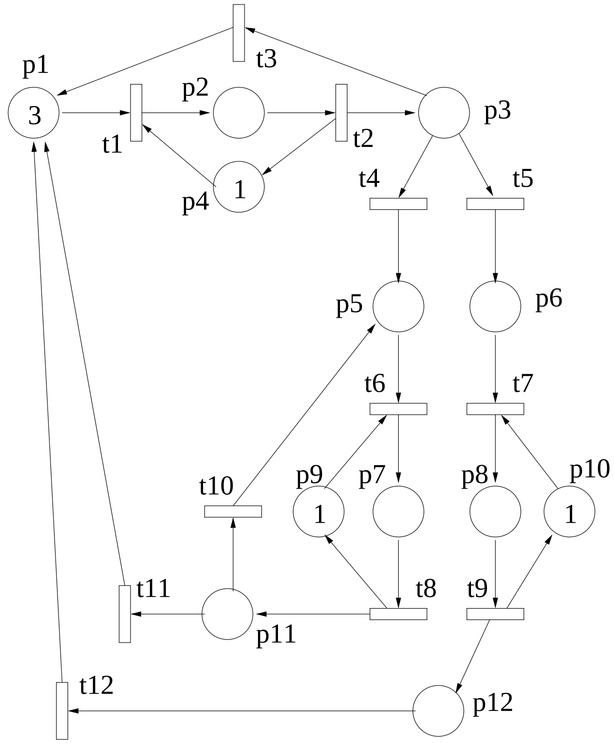


Figure 2.10: Continuous mono-T-semiflow reducible net system, adapted from a queuing network in [18].

Remark 2.29. *Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow reducible net system, and assume $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ keep the same proportion in continuous equal conflicts (i.e., for every pair t_i, t_j in CEQ relation $\boldsymbol{\lambda_1}[t_i]/\boldsymbol{\lambda_1}[t_j] = \boldsymbol{\lambda_2}[t_i]/\boldsymbol{\lambda_2}[t_j]$). The timed contPN systems $\langle \mathcal{N}, \boldsymbol{\lambda_1}, \boldsymbol{m}_0 \rangle$ and $\langle \mathcal{N}, \boldsymbol{\lambda_2}, \boldsymbol{m}_0 \rangle$ have the same visit ratio vector.*

A mono-T-semiflow reducible net can be transformed in a mono-T-semiflow net fusing the transitions in CEQ. The procedure will be discussed in Section 3.6.

In section 2.3.4 a branch and bound algorithm is used to compute upper bounds of the steady state throughput for continuous systems under infinite server semantics, each node corresponding to a LPP. In the case of mono-T-semiflow reducible nets, the bounds can be computed solving only one simple LPP. The linear programming problem is the "continuous version" of the bounds obtained in [17] for discrete nets[1].

Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow reducible net and γ_i the solution of the linear programming problem:

$$\gamma_i = \max\left\{\boldsymbol{y} \cdot \boldsymbol{PD}_i | \boldsymbol{y} \cdot \boldsymbol{C} = 0, \boldsymbol{y} \cdot \boldsymbol{m}_0 = 1, \boldsymbol{y} \geq 0\right\} \tag{2.21}$$

where $\boldsymbol{PD}(p) = \max\limits_{t \in p^\bullet} \frac{\boldsymbol{Pre}[p,t] \cdot \boldsymbol{v}^{(i)}[t]}{\boldsymbol{\lambda}[t]}$ and $\boldsymbol{v}^{(i)}$ is the visit ratio vector normalized for transition t_i.

According to [42] and [17] the throughput of the discrete system in steady state (χ) and the flow of the continuous system under infinite server semantics ($\boldsymbol{f}$) verify $\chi \leq \frac{1}{\gamma} \cdot \boldsymbol{v}^{(i)}$ and $\boldsymbol{f} \leq \frac{1}{\gamma} \boldsymbol{v}^{(i)}$ respectively. Moreover, this bound is reached in the continuous system (i.e. $\boldsymbol{f} =$

[1] In [17] the mono-T-semiflow reducible nets are called Free Related T-semiflows nets (FRT)

$\frac{1}{\gamma}\boldsymbol{v}^{(i)}$) iff the steady state configuration contains the support of a P-semiflow [42]. Generalizing for every possible equilibrium configuration:

Proposition 2.30. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow reducible net system. If every* possible equilibrium configuration *contains the support of a P-semiflow, the throughput of the discrete PN is upper bounded by the throughput of the continuous PN under infinite server semantics.*

Proof. If every possible equilibrium configuration contains the support of a P-semiflow then the steady state configuration contains the support of a P-semiflow and applying Proposition 5 in [42], $\frac{1}{\gamma}\boldsymbol{v}^{(i)}$ is the flow of the continuous system in steady state. □

Hence, for mono-T-semiflow reducible nets under infinite server semantics, the throughput of the system will be given by the slowest P-semiflow when the possible equilibrium configurations contain the support of a P-semiflow.

Chapter 3

Performance properties and simulation

Summary

In the first part of this Chapter we limit our consideration to the class of mono-T-semiflow reducible nets. First it is proved that under some "broad" conditions, infinite server semantics is always a better approximation of the discrete system than finite server semantics, motivating the choice of this semantics in the most part of the thesis. Then, for the same class, monotonicity of the throughput in steady-state with respect to the firing rate and the initial marking, is studied and it is shown under which conditions holds. In the second part of the Chapter, contPNs under infinite server semantics containing immediate transitions are under study. An algorithm to simulate these systems and some techniques to improve simulation and analysis are presented.

3.1 Comparison of server semantics

In this section the throughput of mono-T-semiflow reducible nets under infinite and finite server semantics will be compared. According to Proposition 2.30, the throughput of the markovian discrete net is upper bounded by the throughput of the continuous net system with infinite server semantics under some conditions. Here, we prove that the throughput of continuous net systems under finite server semantics is greater than the throughput under infinite server semantics and conclude that for this class infinite server semantics provides a more accurate approximation of discrete models. The servers are made explicit under continuous infinite server semantics when they are not implicit in the model because otherwise the comparison is inappropriate. This fact ensures that the throughput of the continuous system under both semantics is upper bounded by the same value.

The comparison between contPN systems under finite and infinite server semantics is done under the liveness hypothesis of the untimed contPN system. Liveness is used to ensure that in every moment the continuous system under finite server semantics has at least one strongly-enabled transition (liveness is equivalent to deadlock-freeness for this class). Otherwise, according to the flow definition (2.7) every transition has at least one empty input place and the net is not live as untimed. Liveness analysis of autonomous and timed mono-T-semiflow reducible nets is studied in [43]. A necessary condition for the existence of a marking that makes the system lim-live is given, which can be checked in polynomial time: every transition has at least one place that is not input of any other transition. In [38] it is proved that deadlock-freeness is decidible in general nets, and an algorithm is provided.

Proposition 3.1. *Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow reducible contPN system. For any $\boldsymbol{\lambda}$, the flow in steady state under finite server semantics is greater than or equal to the flow under infinite server semantics.*

Proof. The net is mono-T-semiflow reducible so the throughput in steady state will be proportional to the visit ratio vector for both finite and infinite server semantics ($\boldsymbol{f_F} = \alpha_F \cdot \boldsymbol{v}^{(i)}$, $\boldsymbol{f}_\infty = \alpha_\infty \cdot \boldsymbol{v}^{(i)}$). Under finite server semantics at least one transition will be strongly enabled in steady state (otherwise the net is not live). Let t_i be one of those transitions, and assume that we normalize the visit ratio with respect to this transition. Then $\alpha_F = n_i \cdot \boldsymbol{\lambda}[t_i]$ where n_i is the number of servers for transition t_i. Under infinite server, if $\boldsymbol{m}$ is the steady state marking, it verifies that $\alpha_\infty = \min_{p \in {}^\bullet t_i} \frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p,t_i]} \cdot \boldsymbol{\lambda}[t_i] \leq \min_{p \in {}^\bullet t_i} \boldsymbol{m}[p] \cdot \boldsymbol{\lambda}[t_i]$ ($\boldsymbol{Pre}[p,t_i] \in \mathbb{N}$). Since the place that models the servers of t_i is input of the transition, $\min_{p \in {}^\bullet t_i} \boldsymbol{m}[p] \leq n_i$ and so $\alpha_\infty \cdot \boldsymbol{v}^{(i)} = \boldsymbol{f}_\infty[t_i] \leq \boldsymbol{f_F}[t_i] = \alpha_F \cdot \boldsymbol{v}^{(i)}$. □

Putting together Propositions 2.30 and 3.1 we can conclude that infinite server semantics provides a better approximation of the throughput of the discrete system than finite server semantics, under the expresed conditions.

Theorem 3.2. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow reducible Petri net system with every* possible equilibrium configuration *containing the support of a P-semiflow. For any T-timed interpretation of a discrete model, the continuous model under infinite server semantics provides a better approximation than the continuous model under finite server semantics.*

Example 3.3. *Let us consider again the queuing network presented in Fig. 2.10. We have computed simulations for the discrete stochastic model (exponential distribution for servers) and* *the continuous model under infinite and finite (single-server) server semantics, using* $\boldsymbol{\lambda} = \mathbf{1}$. *The servers were made explicit in the model and are not represented in the figure to simplify it. Every configuration contains the support of a P-semiflow (this will be proved in Example 3.16), so infinite server semantics will fit better. Simulating the model and measuring the flow of transition* t_1 *(the completed flow vector is computed as* $f_1 \cdot \boldsymbol{v}^{(1)}$*, where* $\boldsymbol{v}^{(1)}$ *is given in Example 2.28), the following results are obtained: the throughput is* 0.1337 *for the discrete model,* 0.1667 *for the continuous model under infinite server semantics, and* 0.3333 *for the continuous model under finite server semantics (i.e. two times bigger).*

In general, proving that every configuration contains the support of a P-semiflow is difficult since the number of configurations may be large. However, there are net subclasses for which it is immediate. Take for example EQ nets. A bounded lim-live contPN system is structurally live and structurally bounded as discrete [64], and, for EQ nets this implies that every configuration contains a P-semiflow [76]. Therefore the conditions of Theorem 3.2 are satisfied. Moreover, for this class being structurally live and bounded is equivalent to being conservative, consistent and the rank of the token flow matrix equal to the number of equal conflicts minus one [76]). Hence it can be verified in polynomial time. Moreover, if the initial marking is such that every P-semiflow is marked, the structurally live and structurally bounded EQ net is live as continuous.

Corollary 3.4. *Let* $\mathcal{N}$ *be an EQ net system that is structurally live and structurally bounded as discrete. For any* $\boldsymbol{m}_0$ *such that every P-semiflow is marked, the continuous system* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ *with infinite server semantics provides a better approximation of the throughput of the discrete system than finite server semantics.*

3.2 Monotonicity and fluidification

In this section we will put our attention on the timed conPN system under infinite server semantics and study some properties related to monotonicity. From the flow definition in (2.9) it is easy to observe that if the vector $\boldsymbol{\lambda}$ is multiplied by a constant $k > 0$ then at any reachable marking the flow will be also multiplied by k. When the initial marking of the net system is multiplied by k, the system will be k times faster. But, what happens if only some components of $\boldsymbol{\lambda}$ or only some components of $\boldsymbol{m}_0$ are increased? In general, as happens for discrete nets, increasing the rate of a transition or the initial marking of a place may lead to a slower system (see examples in Subsection 2.2 in [42]). However, this unexpected behavior is usually not desirable in many kinds of systems. For example, replacing a machine by a faster one or adding new machines in a production system should not decrease the throughput. In this section we will see that mono-T-semiflow reducible nets, under quite general conditions, have the a priori expected monotonicity property.

Theorem 3.5. *Let* $\langle \mathcal{N}, \boldsymbol{\lambda_1}, \boldsymbol{m}_0 \rangle$ *and* $\langle \mathcal{N}, \boldsymbol{\lambda_2}, \boldsymbol{m}_0 \rangle$ *be mono-T-semiflow reducible contPN systems and* $\boldsymbol{\lambda_1} \le \boldsymbol{\lambda_2}$ *imposing the same flow proportions in continuous equal conflicts (i.e., for every pair* t_i, t_j *in CEQ relation,* $\boldsymbol{\lambda_1}[t_i]/\boldsymbol{\lambda_1}[t_j] = \boldsymbol{\lambda_2}[t_i]/\boldsymbol{\lambda_2}[t_j]$*). If both systems reach some steady states and both steady state configurations contain the support of a P-semiflow then steady state flows verify* $\boldsymbol{f_1} \le \boldsymbol{f_2}$.

Proof. For $i = 1,2$, let $\boldsymbol{m_i}$ be the steady state marking of $\langle \mathcal{N}, \boldsymbol{\lambda_i}, \boldsymbol{m}_0 \rangle$, and $\boldsymbol{y_i}$ a P-semiflow whose support is contained in one configuration defined by $\boldsymbol{m_i}$.

Let us focus on $\boldsymbol{f_2}$ and $\boldsymbol{y_2}$. Every place $p_{j2} \in ||\boldsymbol{y_2}||$ restricts the flow of at least one of its one output transitions, denoted by t_{j2}, i.e.:

$$\boldsymbol{f_2}[t_{j2}] = \boldsymbol{\lambda_2}[t_{j2}] \cdot \frac{\boldsymbol{m_2}[p_{j2}]}{\boldsymbol{Pre}[p_{j2}, t_{j2}]}$$

Using the P-semiflow $\boldsymbol{y_2}$, we can write the token conservation law for $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$, and taking $\boldsymbol{m_2}$ from the previous equation:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_2}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} = \sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}]$$

Now, $\boldsymbol{m_1}[p_{j2}]$ can be replaced using $\boldsymbol{f_1}$ because

$$\boldsymbol{f_1}[t_{j2}] = \boldsymbol{\lambda_1}[t_{j2}] \cdot enab(t_{j2}, \boldsymbol{m_1}) \leq \boldsymbol{\lambda_1}[t_{j2}] \cdot \frac{\boldsymbol{m_1}[p_{j2}]}{\boldsymbol{Pre}[p_{j2}, t_{j2}]}.$$

Moreover, $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, so:

$$\begin{aligned} &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}] \geq \\ \geq &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_1}[t_{j2}]} \geq \\ \geq &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} \end{aligned}$$

The net is mono-T-semiflow reducible, and $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ keep the same proportion in continuous equal conflicts. Hence, both visit ratios will be the same, $\boldsymbol{v}^{(1)} > 0$. Let $\boldsymbol{f_1} = k_1 \cdot \boldsymbol{v}^{(1)}$ and $\boldsymbol{f_2} = k_2 \cdot \boldsymbol{v}^{(1)}$. Therefore, merging the last equations:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{v}^{(1)}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} \cdot (k_2 - k_1) \geq 0$$

And so $k_2 \geq k_1$. □

The previous result can be extended to sets of rates.

Definition 3.6. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a timed contPN system and let $\mathcal{L} \subseteq \mathbb{R}^{|T|}_{>0}$. The system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ is said to be* monotone for t_j in steady state w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$ *if $\forall \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2} \in \mathcal{L}$ with $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, if these systems have steady states, the associated steady state flows verify $\boldsymbol{f_1}[t_j] \leq \boldsymbol{f_2}[t_j]$.*

In the case of mono-T-semiflow reducible net systems, since the steady state flow is proportional to a vector defined by the net structure and routing (speeds at CEQs), if a system is monotone for a given t_j, it is monotone for $\forall t \in T$. Therefore, it can be said that the net system is monotone (i.e. it is monotone for all transitions).

The previous theorem provides sufficient conditions for a timed system to be monotone w.r.t. $\boldsymbol{\lambda}$.

Theorem 3.7. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}\rangle$ be a timed mono-T-semiflow reducible contPN system under infinite server semantics. Let $\mathcal{L} \subseteq \mathbb{R}^{|T|}_{>0}$ be such that $\forall \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_1 \in \mathcal{L}$ they impose the same flow proportions in continuous equal conflicts (i.e. for every pair t_i, t_j in CEQ relation $\boldsymbol{\lambda_1}[t_i]/\boldsymbol{\lambda_1}[t_j] = \boldsymbol{\lambda_2}[t_i]/\boldsymbol{\lambda_2}[t_j]$). If the* possible equilibrium configurations *of $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}\rangle$ contain the support of a P-semiflow then the timed system is monotone in steady state with respect to $\boldsymbol{\lambda} \in \mathcal{L}$.*

Example 3.8. *Let us consider the contPN in Figure 2.4. This net is mono-T-semiflow with $\boldsymbol{x} = [1,1,1]$ and has 4 configurations: $\mathcal{C}_1 = \{p_1, p_2, p_3\}$, $\mathcal{C}_2 = \{p_1, p_4, p_3\}$, $\mathcal{C}_3 = \{p_4, p_2, p_3\}$ and $\mathcal{C}_4 = \{p_4, p_3\}$, and two minimal P-semiflows: $\boldsymbol{y_1} = p_1 + p_2 + p_3$ and $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$. Therefore there are two configurations that contain a P-semiflow, $\mathcal{C}_1$ and $\mathcal{C}_2$, one configuration that contains the support of a P-flow, $\mathcal{C}_3$ ($\boldsymbol{y_3} = \boldsymbol{y_1} - \boldsymbol{y_2}$) but no P-semiflow, and one configuration that does not contain the support of any P-flow, $\mathcal{C}_4$.*

1) Let $\boldsymbol{m}_0 = [1,1,0,15]$. We will see that the only possible equilibrium configurations for $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ are $\mathcal{C}_1$ and $\mathcal{C}_2$. Both contain P-semiflows, so using Theorem 3.7 the timed system will be monotone with respect to $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ (for this initial marking).

First, let us see that $\mathcal{C}_3$ is not a possible equilibrium configuration. To be an equilibrium configuration, the following system of equations should be satisfied:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 16 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(d) } f_2 \text{ restricted by } p_2 \end{cases} \tag{3.1}$$

where m_1, m_2, m_3, m_4 and f_1, f_2, f_3 is a possible steady state in $\mathcal{C}_3$. From (3.1.a) $\Longrightarrow m_1 \leq 2$ and $m_3 \leq 2$. Moreover, (3.1.c) implies $m_4 \leq m_1 \leq 2$. Therefore, $m_1 + 4 \cdot m_3 + m_4 \leq 2 + 8 + 2 = 12$ and so (3.1.b) cannot be satisfied.

Assume now that $\mathcal{C}_4$ is the steady state configuration, therefore the following system should be satisfied:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 16 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_2 \geq m_4 & \text{(d) } f_2 \text{ restricted by } p_4 \end{cases} \tag{3.2}$$

Observe that the only difference with respect to (3.1) *is the constraint (d). Notice that (d) has not been used above to prove that* (3.1) *cannot have a solution. Therefore* (3.2) *has no solution and $\mathcal{C}_4$ cannot be an equilibrium configuration.*

Thus for $\boldsymbol{m}_0 = [1,1,0,15]^T$ the only possible equilibrium configurations, for any $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$, are $\mathcal{C}_1$ and $\mathcal{C}_2$. Both contain the support of a P-semiflow, thus the timed system is monotone w.r.t. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ (Theorem 3.7). In Figure 3.1 it is shown the evolution of the throughput of the net for $\boldsymbol{m}_0 = [1,1,0,15]^T$ and $\boldsymbol{\lambda} = [1,z,1]^T$ with $0 < z \leq 5$ and, as expected, the throughput of the system increases when λ_2 increases.

2) Let $\boldsymbol{m}_0 = [15,1,1,0]$. For this marking, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$ can be equilibrium configurations choosing appropriate values for $\boldsymbol{\lambda}$. For example: $\boldsymbol{m_2} = [1,13,3,6]^T$ in $\mathcal{C}_2$ is an equilibrium marking for $\boldsymbol{\lambda_2} = [6,1,2]^T$ with $\boldsymbol{f_2} = \mathbf{6}$; $\boldsymbol{m_3} = [13.67,3,0.33,4]^T$ in $\mathcal{C}_3$ is an equilibrium marking for $\boldsymbol{\lambda_3} = [3,2,18]^T$ with $\boldsymbol{f_3} = \mathbf{1}$; $\boldsymbol{m_4} = [10.5,4.5,2,0.5]^T$ in $\mathcal{C}_4$ is an equilibrium marking for $\boldsymbol{\lambda_4} = [8,4,1]^T$ with $\boldsymbol{f_4} = \mathbf{2}$. Since some possible equilibrium configurations do not contain the support of a P-semiflow, monotonicity cannot be guaranteed.

In Figure 3.2 it is sketched the evolution of the throughput of the net system for $\boldsymbol{m}_0 = [15,1,1,0]^T$ and $\boldsymbol{\lambda} = [1,z,1]^T$ with $0 < z \leq 5$. It can be seen that it is not monotonic, even a

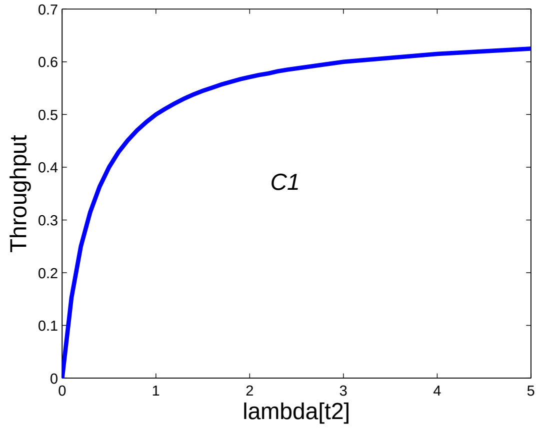


Figure 3.1: Throughput of the contPN system in fig. 2.4 for $\boldsymbol{m}_0 = [1,1,0,15]^T$ and different values of λ_2.

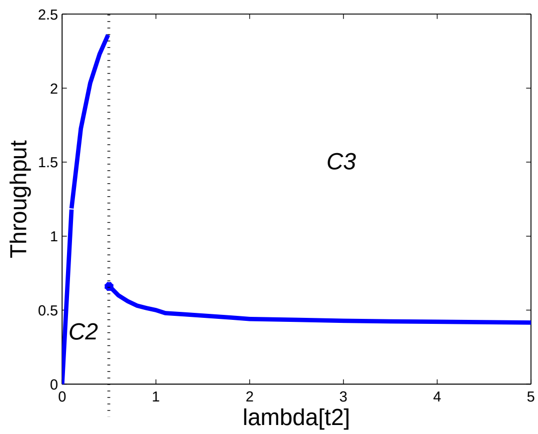


Figure 3.2: Throughput of the contPN system in fig. 2.4 for $\boldsymbol{m}_0 = [15,1,1,0]^T$ and different values of λ_2.

discontinuity exists at $\lambda_2 = 0.5$. When $0 < \lambda_2 < 0.5$, the equilibrium configuration is $\mathscr{C}_2$ and the throughput is increasing. For $\lambda_2 \geq 0.5$ the steady state configuration becomes $\mathscr{C}_3$ that contains the support of a P-flow and the steady state throughput is decreasing (the conditions of Theorem 3.7 are not satisfied). Therefore, for $\boldsymbol{m}_0 = [15,1,1,0]^T$ the system is not monotone in steady state w.r.t. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$.

3) Let us now study the monotonicity of the contPN with $\boldsymbol{m}_0 = [15,1,1,0]^T$ w.r.t $\boldsymbol{\lambda} \in \mathscr{L}_1$ where $\mathscr{L}_1 = \{[\lambda_1,\lambda_2,\lambda_3]^T | \lambda_1 = \lambda_3 = 1, 0 < \lambda_2 < 0.5\}$.

First, $\mathscr{C}_4$ cannot be an equilibrium configuration. If it were, p_4 restricts the flows of t_1 and t_2 in steady state, and so since the steady state flow verifies $f_1 = f_2 = f_3$, $\lambda_2 \cdot m_4 = \lambda_1 \cdot \frac{m_4}{2} = \frac{m_4}{2}$. We are assuming $\lambda_2 < 0.5$, therefore $m_4 = 0$ and $f_1 = f_2 = f_3 = 0$, i.e., the system is in a deadlock. Notice that f_3 is defined by m_3, so $m_3 = 0$. Using the P-semiflows the following equations are obtained: $m_1 + m_2 = 17$ and $m_1 = 19$ which cannot be satisfied.

If $\mathscr{C}_3$ were the equilibrium configuration, since the steady state flow verifies $f_1 = f_2 = f_3$, *then $\lambda_1 \cdot \frac{m_4}{2} = \lambda_2 \cdot m_2 = \lambda_3 \cdot m_3$. But $\lambda_1 = \lambda_3 = 1$, and so $\frac{m_4}{2} = \lambda_2 \cdot m_2 = m_3$. Hence, the following system of equations should have a solution:*

$$\begin{cases} m_1 + m_2 + m_3 = 17 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 19 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(d) } f_2 \text{ restricted by } p_2 \\ \frac{m_4}{2} = \lambda_2 \cdot m_2 = m_3 & \text{(e) steady state flows} \end{cases} \tag{3.3}$$

Using (3.3.e), $m_4 = 2\lambda_2 \cdot m_2$ and considering (3.3.d) the following inequality is obtained: $2\lambda_2 \cdot m_2 \geq m_2$. But $m_2 > 0$ because the system cannot deadlock (see the reasoning for $\mathscr{C}_4$) therefore $2 \cdot \lambda_2 \geq 1$ that cannot be satisfied since $0 < \lambda_2 < 0.5$. Hence, the time contPN system is monotone w.r.t. $\boldsymbol{\lambda} \in \mathscr{L}_1$ since the only possible equilibrium configurations are $\mathscr{C}_1$ and $\mathscr{C}_2$ that contain the support of a P-semiflow.

Theorem 3.9. *Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a mono-T-semiflow reducible contPN under infinite server semantics, and let $\boldsymbol{m_1} \leq \boldsymbol{m_2}$. If for every $i = 1,2$ $\langle \mathcal{N}, \lambda, \boldsymbol{m_i} \rangle$ reaches a steady state and the steady state configuration contains the support of a P-semiflow, then the steady state flows verify $\boldsymbol{f_1} \leq \boldsymbol{f_2}$.*

Proof. According to [42], $\boldsymbol{f_i} = \frac{1}{\gamma_i} \cdot \boldsymbol{v}^{(1)}$ with $\gamma_i = \max\left\{\boldsymbol{y} \cdot \boldsymbol{PD} | \boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}, \boldsymbol{y} \cdot \boldsymbol{m_i} = \boldsymbol{1}, \boldsymbol{y} \geq \boldsymbol{0}\right\} = \max\left\{\frac{1}{\boldsymbol{y} \cdot \boldsymbol{m_i}} \cdot \boldsymbol{y} \cdot \boldsymbol{PD} | \boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0}\right\}$. Let us assume $\boldsymbol{m_{10}} \leq \boldsymbol{m_{20}}$. Then, for every P-semiflow $\boldsymbol{y}$, $\boldsymbol{y} \cdot \boldsymbol{m_1} \leq \boldsymbol{y} \cdot \boldsymbol{m_2}$ and so, $\gamma_1 \geq \gamma_2$. Therefore, $\boldsymbol{f_1} \leq \boldsymbol{f_2}$. ☐

Monotonicity w.r.t. the initial marking can be studied also on a set of markings as done in the case of the monotonicity w.r.t. $\boldsymbol{\lambda}$.

Definition 3.10. *Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a timed contPN and let $\mathscr{M} \subseteq \mathbb{R}^{|P|}_{>0}$. The system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ is said to be* monotone for t_j in steady state w.r.t. $\boldsymbol{m}_0 \in \mathscr{M}$ *if $\forall \boldsymbol{m}_{01}, \boldsymbol{m}_{02} \in \mathscr{M}$ with $\boldsymbol{m}_{01} \leq \boldsymbol{m}_{02}$, if these systems have steady states, the associated steady state flows verify $\boldsymbol{f_1}[t_j] \leq \boldsymbol{f_2}[t_j]$.*

As in the case of the monotonicity w.r.t. $\boldsymbol{\lambda}$, Theorem 3.9 can be used to derive the following sufficient condition for monotonicity of a mono-T-semiflow net system w.r.t. the initial marking.

Theorem 3.11. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed mono-T-semiflow reducible contPN system under infinite server semantics. If for every $\boldsymbol{m}_0 \in \mathscr{M}$ the* possible equilibrium configurations *of $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ contain the support of a P-semiflow then $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is monotone in steady state w.r.t. $\boldsymbol{m}_0 \in \mathscr{M}$.*

Example 3.12. *Let us consider the contPN in figure 2.4. Assume now that $\boldsymbol{\lambda} = [1,1,1]^T$.*

1) Let us study the monotonicity w.r.t. $\boldsymbol{m}_0 \in \mathscr{M}_1 = \{[m_1, m_2, m_3, m_4]^T | m_1 = 0, m_2 = m_3 = 1, m_4 > 0\}$.

$\mathscr{C}_4$ cannot be an equilibrium configuration. If it were, in steady state p_4 would limit the flow of t_1 and t_2 and taking into account the T-semiflow: $f_1 = f_2 \to \frac{m_4}{2} = m_4$ (because $\lambda_1 = \lambda_2 = 1$). Then the only solution is $m_4 = 0$ and from the T-semiflow, $m_3 = 0$. Writing the P-semiflows we have: $m_1 + m_2 = 2$ and $m_1 = 4 + z$, with z the initial marking of p_4. Since $z \geq 0$ these equations cannot be satisfied. Therefore $\mathscr{C}_4$ cannot be an equilibrium configuration.

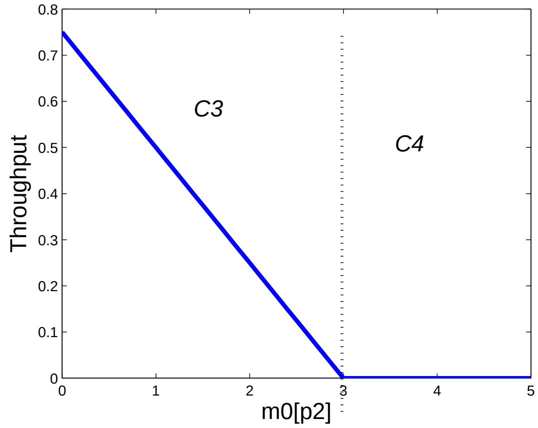


Figure 3.3: Throughput of the contPN system in fig. 2.4 for $\boldsymbol{m}_0 = [15, z, 1, 0]^T$ and $\boldsymbol{\lambda} = [1, 1, 1]^T$.

For $\mathscr{C}_3$, the following system of equations should have a solution:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4m_3 + m_4 = 4 + z & \text{(b) second P-semiflow} \\ \frac{m_4}{2} = m_2 = m_3 & \text{(c) steady state flow} \\ m_1 \geq m_4 & \text{(d) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(e) } f_2 \text{ restricted by } p_2 \end{cases} \tag{3.4}$$

where z is the initial marking of p_4. From (3.4.b) - (3.4.a) we obtain: $3 \cdot m_3 - m_2 + m_4 = z + 2$ and replacing m_3 and m_4 using (3.4.c), $3 \cdot m_2 - m_2 + 2 \cdot m_2 = z + 2$ and so $m_2 = \frac{z+2}{4}$. Hence, any solution should be of the form: $[m_1, \frac{z+2}{4}, \frac{z+2}{4}, \frac{z+2}{2}]^T$ with $m_1 \geq \frac{z+2}{2}$ (3.4.d). But (3.4.a) implies: $m_1 = 2 - \frac{z+2}{4} - \frac{z+2}{4} = \frac{-z+2}{2}$ and combining with (3.4.d) results $\frac{-z+2}{2} \geq \frac{z+2}{2} \rightarrow 2 \cdot z \leq 0 \rightarrow z \leq 0$ which is impossible because z is the initial marking of p_4.

Thus neither $\mathscr{C}_4$ nor $\mathscr{C}_3$ can be equilibrium configurations. Using Theorem 3.11 the time net system is monotone w.r.t. the initial marking $\boldsymbol{m}_0$ in $\mathscr{M}_1$.

2) Let us consider $\mathscr{M}_2 = \{[m_1, m_2, m_3, m_4]^T | m_1 = 15, m_3 = 1, m_4 = 0, m_2 > 0\}$. $\mathscr{C}_3$ and $\mathscr{C}_4$ can be equilibrium configurations and monotonicity can be lost. Indeed, simulating for $\boldsymbol{m}_0[p_2] \in [0, 5]$ (Figure 3.3), the throughput decreases, even deadlock is reached for $\boldsymbol{m}_0[p_2] \geq 3$. Hence the timed net system with $\boldsymbol{m}_0 \in \mathscr{M}_2$ is not monotonic in steady state.

3.3 Some properties of non-monotonicity

As an immediate consequence of Theorems 3.7 and 3.11, if all configurations defined by $\mathscr{N}$ (i.e., independently of $\boldsymbol{m}_0$), contain a P-semiflow, then the underlying net system is monotone in steady state w.r.t. the set of $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{\geq 0}$ that impose the same routing, and w.r.t. $\boldsymbol{m}_0$ in $\mathbb{R}^{|P|}_{\geq 0}$. Moreover, we will prove that this P-semiflow condition when asked to all the configurations is in fact equivalent to an analogous P-flows condition (Corollary 3.15). Let us first consider the following Lemma.

Lemma 3.13. *Let $\mathscr{N}$ be a consistent join free net (i.e., $\forall\, t \in T$, $|^\bullet t| \leq 1$). For every P-flow $\boldsymbol{y}$ there exist a P-semiflow $\boldsymbol{y}'$ such that $||\boldsymbol{y}'|| \subseteq ||\boldsymbol{y}||$.*

Proof. Dual of Theorem 9 of [75] (T-flows of CF nets). □

Theorem 3.14. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow reducible contPN system under infinite server semantics. If there exist $\mathcal{L}$ (keeping the rates in the conflicts) or $\mathcal{M}$ such that $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone in steady state w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$ or w.r.t. $\boldsymbol{m}_0 \in \mathcal{M}$, then there exists a configuration that does not contain any P-flow.*

Proof. If $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone, applying Theorem 3.7 or Theorem 3.11, an equilibrium configuration exists that does not contain a P-semiflow. If this equilibrium configuration does not contain the support of any P-flow this is the requested configuration. Otherwise, assume that this equilibrium configuration contains one P-flow (or more). Let us consider the subnet $\mathcal{N}'$ defined by the set of all transitions together with the places that limit their flow in steady state, and let us call $\boldsymbol{C}'$ the token flow matrix of this subnet. Since the original net is mono-T-semiflow, it is consistent. The same T-semiflow is also a right annuller of $\boldsymbol{C}'$, hence $\mathcal{N}'$ is consistent.

If $\mathcal{N}'$ were a JF net, using Lemma 7.8 the support of a P-semiflow should be included in $\mathcal{N}'$, but that is impossible by assumption. Hence $\mathcal{N}'$ must have at least one join. Let us call t_k this transition. Let p_i and p_j be the input places of t_k that belong to the configuration. Obviously, only one place restricts the flow of t_k. Let us assume p_i to be this one.

If we consider now that t_k is restricted by p_j and the other transitions are restricted by the same places as before, we obtain a new configuration (possibly a non-equilibrium one) in which place p_i has been removed. If this configuration contains a P-flow, the reason can be repeated and other place can be removed. At the end, this procedure will define a configuration that does not contain any P-semiflow (from hypothesis) or P-flow. □

Therefore:

Corollary 3.15. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a continuous mono-T-semiflow reducible net under infinite server semantics. If all the configurations contain the support of a P-flow, then the underlying net system is monotonic in steady state w.r.t. $\boldsymbol{m}_0 \in \mathcal{M}$ $\forall \mathcal{M} \subseteq \mathbb{R}^{|P|}_{>0}$, and w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$, $\forall \mathcal{L} \subseteq \mathbb{R}^{|T|}_{>0}$ that imposes the same flow proportions to continuous equal conflicts.*

Example 3.16. *Let us consider again the queuing network presented in Fig. 2.10. It has the following P-semiflows: $\boldsymbol{y_1} = p_2 + p_4$, $\boldsymbol{y_2} = p_7 + p_9$, $\boldsymbol{y_3} = p_8 + p_{10}$ and $\boldsymbol{y_4} = p_1 + p_2 + p_3 + p_5 + p_6 + p_7 + p_8 + p_{11} + p_{12}$. We will see that all configurations contain the support of a P-semiflow.*

First observe that p_2, p_3, p_7, p_8, p_{11} and p_{12} belong to all configurations because they are the only input places in their output transitions. Starting with $\boldsymbol{y_1}$, if its support does not belong to the configuration, p_4 cannot restrict the flow of t_2 forcing us to include p_1. For $\boldsymbol{y_3}$, since p_7 belongs to all configurations, p_9 is not taken and to cover t_6 we have to take p_5. Finally, for $\boldsymbol{y_3}$ as p_8 belongs to all configurations, we cannot take p_{10} forcing us to take p_6. So, we are forced to take: p_1, p_2, p_3, p_5, p_6, p_7, p_8, p_{11}, p_{12}, which is the support of $\boldsymbol{y_4}$.

Hence, all configurations contain the support of a P-semiflow and both kinds of monotonicity properties hold.

3.4 Algorithms to check monotonicity

In Section 3.2, monotonicity of the throughput w.r.t. firing speed $\boldsymbol{\lambda}$ or initial marking is proved for mono-T-semiflow reducible nets whose possible equilibrium configurations contain the support of a P-semiflow. Under the same hypothesis, plus liveness, in Section 3.1 it is proved that infinite server semantics provides a better approximation of a discrete model than finite server semantics. Therefore, it is really interesting to know the configurations that do not contain the support of a P-semiflow because if these configurations are possible equilibrium configurations the previous two properties may not hold.

A first idea is to use boolean equations to represent the conditions the places in a configuration have to fulfill. Let γ_i be a boolean variable such that $\gamma_i = 1$ iff p_i belongs to the configuration: (1) any P-semiflow will provide a boolean equation: the product of the boolean variables associated to the support of the P-semiflow should be 0 if the P-semiflow does not belong to the configuration; (2) if a transition is not a join ($|^\bullet t_i| = 1$) its input place must belong to all configurations (it is an essential cover), therefore the boolean variables associated to those places are 1 and can be removed. For every synchronization, we should have another boolean equation ensuring that at least one input place is taken so that the solution is (or contains) a configuration. The solutions of this system of boolean equations provide configurations that do not contain the support of a P-semiflow.

Example 3.17. *Let us consider the net in Fig. 2.4. This net has been used in Examples 3.8 and 3.12 where the configurations that do not contain the support of a P-semiflow were supposed to be known. These configurations can be computed, for example, using the above algorithm. This net has two P-semiflows:* $\boldsymbol{y_1} = p_1 + p_2 + p_3$ *and* $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$*, so their two boolean equations are:* $\gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0$ *and* $\gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0$*. In order to cover transitions we need additional equations. For* t_1*:* $\gamma_1 + \gamma_4 = 1$ *(*m_1 *or* m_4 *limit the flow of* t_1*) and the same for* t_2*:* $\gamma_2 + \gamma_4 = 1$ *(*m_2 *or* m_4 *limit the flow of* t_2*). Clearly,* p_3 *being the only input place in* t_3 *is an essential cover thus* $\gamma_3 = 1$*. The following system of boolean equations is obtained:*

$$\begin{cases} \gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0 & \textit{(a) First P-semiflow not contained} \\ \gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0 & \textit{(b) Second P-semiflow not contained} \\ \gamma_1 + \gamma_4 = 1 & \textit{(c) } t_1 \textit{ is covered} \\ \gamma_2 + \gamma_4 = 1 & \textit{(d) } t_2 \textit{ is covered} \\ \gamma_3 = 1 & \textit{(e) } t_3 \textit{ is covered} \end{cases} \tag{3.5}$$

From (3.5.e), taking into account (3.5.a) and (3.5.b), $\gamma_1 \cdot (\gamma_2 + \gamma_4) = 0$*. Using (3.5.d):* $\gamma_1 = 0$*, thus* $\gamma_4 = 1$ *and* $\gamma_2 = \oslash$*. In summary:* $\gamma_1 = 0$*,* $\gamma_2 = \oslash$*,* $\gamma_3 = 1$ *and* $\gamma_4 = 1$ *telling us that the net has two configurations (*$\mathscr{C}_1 = \{p_3, p_4\}$*,* $\mathscr{C}_2 = \{p_2, p_3, p_4\}$*) that do not contain the support of a P-semiflow. Depending on the initial marking these configurations can or cannot be equilibrium configurations and monotonicity may be lost (see Example 3.8).*

Remark 3.18. *It can be seen in* (3.5) *that for every transition with only one input place (there* $|^\bullet t_3| = |\{p_3\}| = 1$*) a boolean equation* $\gamma_3 = 1$ *is introduced. Before solving the equations, these variables can be removed reducing the number of variables. The interpretation in the PN is that* t_3 *and* p_3 *can be removed in the model because it belongs to all configurations. If all boolean variable corresponding to transitions with only one input place are removed, the order of the system is lower.*

According to Corollary 3.15, if all configurations contain the support of a P-flow, the timed net system is monotonic in steady state w.r.t. $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{>0}$ and w.r.t. $\boldsymbol{\lambda} \in \mathscr{L}$, $\forall \mathscr{L} \subseteq \mathbb{R}^{|T|}_{>0}$ that impose the same flow proportions to continuous equal conflicts. A second algorithm can be used to check if all configurations contain the support of a P-flow.

First, all configurations are computed (which is exponential) and then, each configuration is checked to see whether it contains the support of a P-flow. For the second step, if

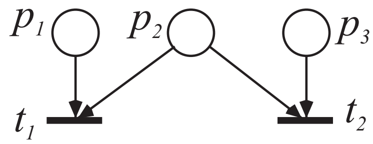


Figure 3.4: Conflict between immediate transitions.

the token flow matrix of a configuration $\mathscr{C}_j$ is denoted by $\boldsymbol{C_j}$ ($\boldsymbol{C_j}$ is obtained from the token flow matrix of the original net removing some rows), to check if $\mathscr{C}_j$ contains the support of a P-flow, is equivalent to check if the following system has a solution (polynomial time complexity): $\exists \boldsymbol{y}$ such that $\boldsymbol{y} \cdot \boldsymbol{C_j} = 0$.

One remark should be done: the second algorithm may fail, but this does not imply that the net system is not monotonic, since the net may include configurations not containing the support of a P-semiflow but which cannot be equilibrium configurations for that initial marking and transition rates (see Example 3.8). In fact, with the second algorithm it is monotonicity for any $\boldsymbol{\lambda} > \boldsymbol{0}$ and $\boldsymbol{m}_0 \geq \boldsymbol{0}$ that is obtained.

Therefore, it seems reasonable first to check with the second algorithm if all configurations contain the support of a P-flow. If not, we can compute configurations not containing the support of a P-semiflow checking for each configuration $\mathscr{C}_j$ with the associated token flow matrix $\boldsymbol{C_j}$ if the following system has a solution: $\exists \boldsymbol{y} \geq 0$, $\boldsymbol{y} \neq 0$ such that $\boldsymbol{y} \cdot \boldsymbol{C_j} = 0$. If we prove that they cannot be equilibrium configurations, the system is monotonic in steady-state.

3.5 Simulation of immediate transitions

As pointed out in section 2.3.2, the simulation of contPNs systems under infinite server semantics containing immediate transitions is a difficult task, as using the flow definition 2.9, their flow result infinite. Therefore, an algorithm is necessary for the computation of the instantaneous marking.

The numerical integration of timed contPN with immediate transitions can be seen as done at two levels: (1) if at least one immediate transition is enabled, stop the flows through all timed transitions and fire immediate transitions until none is enabled; (2) after that, continue computing flows through timed transitions. Numerical techniques developed for ODEs can be applied to the above computational schema if flows through immediate transitions are solved.

For step 1, since all timed transitions are stopped, assume that they are "deleted" from the net. In the remaining model some transitions may be in *conflict*. Compute the coupled conflict sets of the immediate transitions and then, for each coupled conflict set, fire the enabled transitions according to their rates. That is, assume t_1 and t_2 in Fig. 3.4 are immediate transitions with rates $\boldsymbol{r}[t_1] = \alpha$ and $\boldsymbol{r}[t_2] = \beta$. To be consistent with the way immediate transitions are used in discrete nets, t_1 and t_2 should be fired according to their relative rates, until one of the input places goes empty. If it is p_2 that empties first ($\boldsymbol{m}[p_1] \geq \frac{\alpha}{\alpha+\beta} \cdot \boldsymbol{m}[p_2], \boldsymbol{m}[p_3] \geq \frac{\beta}{\alpha+\beta} \cdot \boldsymbol{m}[p_2]$), we are done since t_1 and t_2 are not further enabled. Otherwise, fire the transition that remains enabled until one of its input places is empty.

Hence, at a certain marking $\boldsymbol{m}$, it is possible that not all the transitions in a coupled conflict set are enabled. This introduces a new concept, *effective structural conflict relation* at marking $\boldsymbol{m}$, which is the same as structural conflict relation, but applied to the net without the disabled transitions.

Let $\boldsymbol{q}[t_i]$ be the amount in which t_i is fired. At a marking $\boldsymbol{m}$, the algorithm to be applied to each $\text{CCS}_i \in \text{SCCS}$ is:

Algorithm 3.19. ***procedure*** COMPUTE THE SUBSET OF TRANSITIONS IN CCS_i THAT ARE ENABLED AT THE CURRENT MARKING $\boldsymbol{m}$

Input: $\boldsymbol{m}$, $\text{CCS}_i = \{t_1, \cdots, t_g\}$

Let CI *= the effective coupled conflict sets of* CCS_i *at* $\boldsymbol{m}$

while $CI \neq \emptyset$ ***do***

Let $c = \{t_1, t_2, \ldots, t_k\} \in CI$

Solve the following linear programming problem (LPP):

$$\begin{array}{l}\max \boldsymbol{q}[t_1] + \boldsymbol{q}[t_2] + \cdots + \boldsymbol{q}[t_k] \\ s.t. \\ \begin{cases} \frac{1}{\boldsymbol{r}[t_1]} \cdot \boldsymbol{q}[t_1] = \frac{1}{\boldsymbol{r}[t_2]} \cdot \boldsymbol{q}[t_2] = \cdots = \frac{1}{\boldsymbol{r}[t_k]} \cdot \boldsymbol{q}[t_k] \\ \boldsymbol{m}[p_j] - \sum\limits_{t_h \in p_j^{\bullet}} \boldsymbol{Pre}[p_j, t_h] \cdot \boldsymbol{s}[t_h] \geq 0, \forall p_j \in {}^{\bullet}t_i \\ \boldsymbol{q}[t_1], \boldsymbol{q}[t_2], \cdots, \boldsymbol{q}[t_k] \geq 0 \end{cases}\end{array}$$

Fire $\{t_1, t_2, \cdots, t_k\}$ *with* $\boldsymbol{q}[t_1], \boldsymbol{q}[t_2], \cdots, \boldsymbol{q}[t_k]$ *and let* $\boldsymbol{m}'$ *be the obtained marking, i.e.* $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C} \cdot \boldsymbol{q}$.

Let $CI = (CI \setminus c) \cup \{$ *the effective coupled conflict sets of* c *at* $\boldsymbol{m}'\}$, *and* $\boldsymbol{m} = \boldsymbol{m}'$

end while

end procedure

If the net does not have self-loop arcs associated to immediate transitions, at each execution of the loop at least one of the transitions will not be enabled anymore. Otherwise it may happen that the procedure has to be repeated several times until one place gets empty.

If sequences of several immediate transitions exist, the order in which the elements of the SCCS are visited is important and should be obtained first. It has to ensure that when applying the algorithm to one coupled conflict set, transitions belonging to a previously fired set do not become enabled. This order can be solved as long as we are not dealing with a circuit of immediate transitions, which clearly is a modeling error.

The net in Figure 3.5, taken from [70], models a simple manufacturing cell. The size of its reachability set is not very big and so it could have been analyzed as discrete. However, we have rather kept its token load small so as to observe the quality of the continuous approximation. Notice that the computational effort for the discrete analysis will increase in a more loaded system, while it will not significatively change for the continuous model, and the quality of the approximation usually improves with more loaded systems. To simplify the presentation, only the steady state results will be compared, although the simulation will also give the transient behavior. The throughput of the system as discrete is 0.0412, while it is 0.0415 as continuous. However, the simulation time for the continuous model until the steady state is obtained is 186 seconds using MATLAB in a Pentium IV 3.2 MHz. Quite large for such a simple net. Efficient algorithms for the solution of linear ODEs exist, however the relative abundance of immediate transitions and synchronizations slows down the simula-

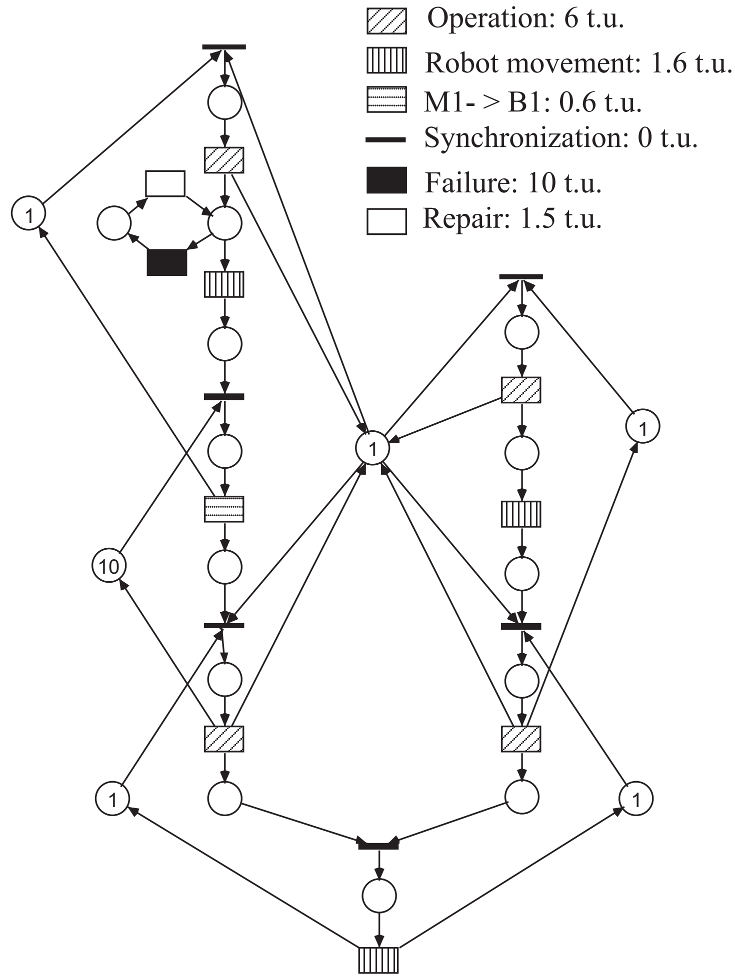


Figure 3.5: Example of a simple manufacturing cell taken from [70].

tion. Hence, if the number of synchronizations and immediate transitions are smaller, the simulation is faster.

3.6 Model reduction

Any set of transitions in CEQ relation can be reduced to a single transition, t. Although the idea of the transformation is the same both for timed and immediate transitions, the new rates and arc weights are not computed the same way. The reason is that in a conflict among immediate transitions, their firing rate does not depend on the enabling, while it does in the timed case. To simplify the presentation here we will consider that the set has only two elements: t_1 and t_2. Then $\forall p \in P$:

- Timed transitions
  - $\boldsymbol{Pre}[p,t] = \boldsymbol{Pre}[p,t_1]$
  - $\boldsymbol{Post}[p,t] = \frac{1}{\boldsymbol{\lambda}[t_1]+\boldsymbol{\lambda}[t_2]} \cdot (\boldsymbol{Post}[p,t_1] \cdot \boldsymbol{\lambda}[t_1] + \boldsymbol{Post}[p,t_2] \cdot \boldsymbol{\lambda}[t_2] \cdot \frac{\boldsymbol{Pre}[p,t_1]}{\boldsymbol{Pre}[p,t_2]})$

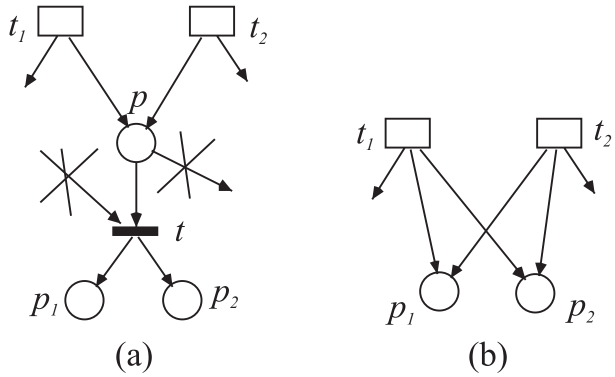


Figure 3.6: Removing a non-synchronizing immediate transition that is not in conflict relation with any other transition.

  - $\boldsymbol{\lambda}[t] = \boldsymbol{\lambda}[t_1] + \boldsymbol{\lambda}[t_2]$

- Immediate transitions,
  - $\boldsymbol{Pre}[p,t] = \boldsymbol{Pre}[p,t_1]$,
  - $\boldsymbol{Post}[p,t] = \frac{\boldsymbol{Pre}[p_1,t_1]}{\boldsymbol{r}[t_1]\cdot\boldsymbol{Pre}[p_1,t_1]+\boldsymbol{r}[t_2]\cdot\boldsymbol{Pre}[p_1,t_2]} \cdot (\boldsymbol{r}[t_1]\cdot\boldsymbol{Post}[p,t_1] + \boldsymbol{Post}[p,t_2]\cdot\boldsymbol{r}[t_2])$

This rule can be applied for example to merge transitions t_{OK} and t_{KO} in Fig. 2.7.a. Let us call this transition t_{OK-KO}.

Pre- and post-fusion rules for discrete PNs were first introduced by Berthelot in [12]. In their original form, they were designed for autonomous discrete nets, but they have been extended/refined afterwards for different kinds of timing [56, 73]. Similar rules can also be applied for continuous nets. Fluidification simplifies the problem of weights in the input arcs of a transition, which were difficult to deal with in the discrete case, and also the problem of transitions in a CEQ, since they can be merged using the result in the previous section.

Let p be a place with only one output transition t which does not have other input place. That is ${}^\bullet t = p$ and $p^\bullet = t$, see Fig. 3.6.a. Both p and t can be removed, and the input transitions of p will now split the flow among the output places of t. More formally, let ${}^\bullet p = \{t_1, \cdots t_k\}$, $t^\bullet = \{p_1, \cdots p_r\}$ and $\boldsymbol{m}_0[p] = 0$. In the reduced net, $\boldsymbol{Post}[p_i,t_j] = \boldsymbol{Post}[p,t_j]\cdot \boldsymbol{Post}[p_i,t]/\boldsymbol{Pre}[p,t]$. In the figure the input transitions are represented as timed, but the transformation can be used also with immediate transitions.

This rule allows to remove transitions t_{start}, t_{sync} and t_{OK-KO} in the net in Fig. 2.7.a (recall that place p_5 has been removed before). Hence, in this case all the immediate transitions of the net have been removed.

A similar rule can be obtained changing the role of immediate and timed transitions. On one side this rule is slightly more restrictive because it requires that none of the immediate transitions is in structural conflict relation with any other transition (that is, all the immediate transitions are persistent) and their only output is p. On the other side, it is more general since it allows synchronizations in t (see Fig. 3.7.a). The arc weights are computed as in the previous rule.

Other situation that can be simplified can be seen in Fig. 3.8.a. This kind of situation appears in the example we will study in the following section. In this case, a timed transition is followed by an immediate one. It is not difficult to see that although a synchronization

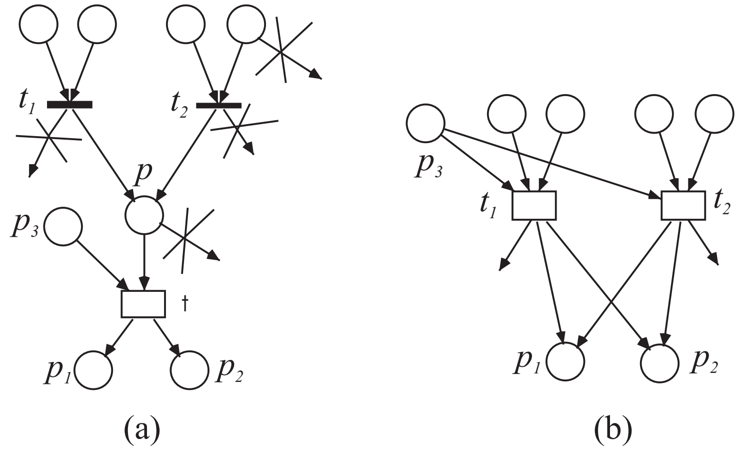


Figure 3.7: A kind of symmetric rule to the one presented in Fig. 3.6.

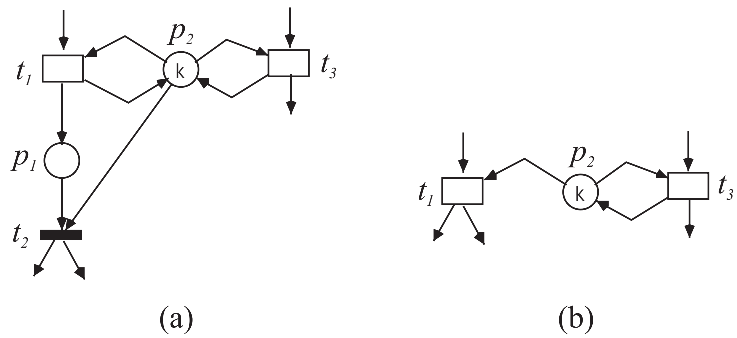


Figure 3.8: All the tokens that arrive to place p_1 will immediately go through.

appears at the immediate transition, it will never stop the flow that comes from the input transition as long as no other transition consumes tokens from the place represented as p_2. That is, p_2 could have other output arcs if they are equal-weighted self-loops. Then, (p_2, t_2) is implicit and so, transition t_2 will fire as soon as tokens arrive to p_1, and its firing takes no time. Hence, both transitions can be fused. Observe that this could not have been done if both were timed, or if t_1 were immediate and t_2 were timed.

3.7 Case studies

3.7.1 Example 1

Let us consider the open markovian queuing network in Figure 3.9, consisting in three nodes with infinite capacity queues, exponentially distributed service time ($\mu_1 = \mu_2 = 4$ and $\mu_3 = 2$) for the servers, a single Poisson arrival process with rate $r_1 = 1$ and feedback paths with given routing probabilities ($pr_{11} = 30\%$, $pr_{12} = 20\%$, $pr_{13} = 50\%$, $pr_{21} = 20\%$ and $pr_{22} = 80\%$).

The PN model corresponding to this queuing network is sketched in Figure 3.10, for which: transition IN models the arrival process of the clients and has $\boldsymbol{\lambda}[\boldsymbol{IN}] = r_1 = 1$; m_1, m_2, m_3 model the serving process by each server; t_{11}, t_{12} and t_{13} have associated the proba-

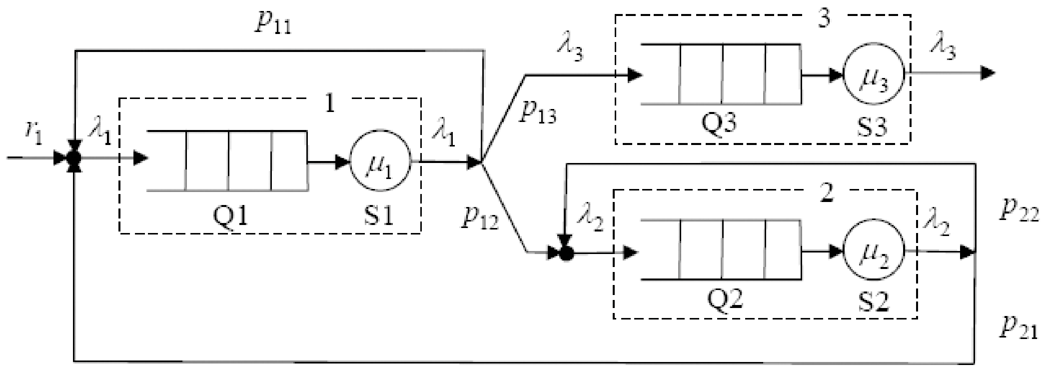


Figure 3.9: Structure of the open queueing network used in Example 3.7.1.

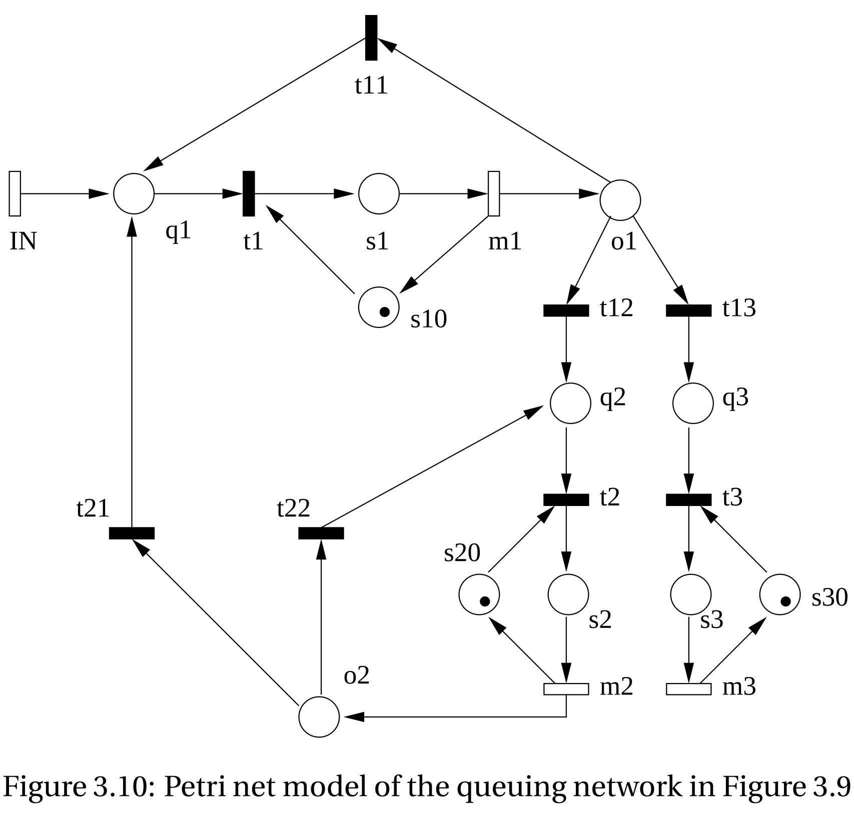


Figure 3.10: Petri net model of the queuing network in Figure 3.9.

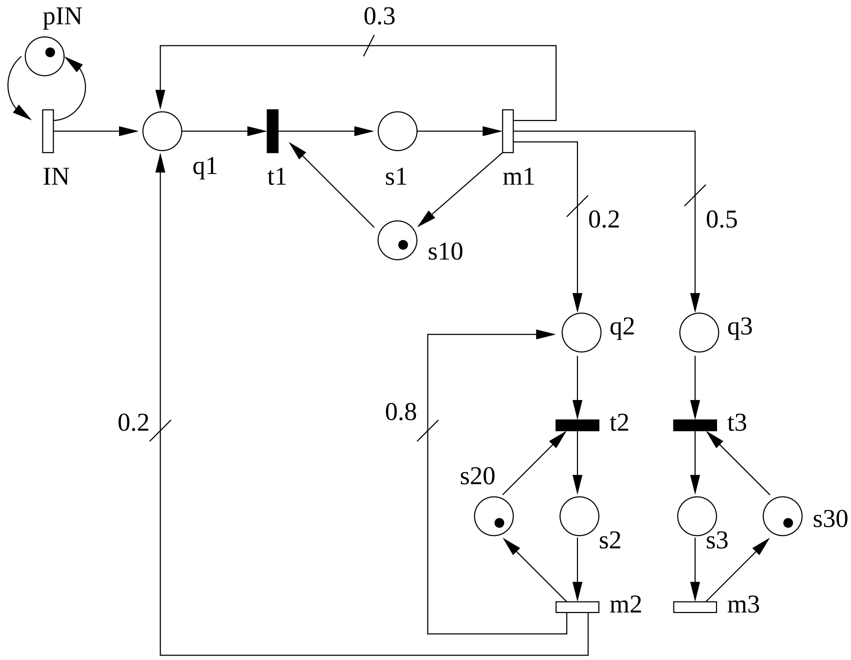


Figure 3.11: Equivalent Petri net model of the queuing network in Figure 3.9.

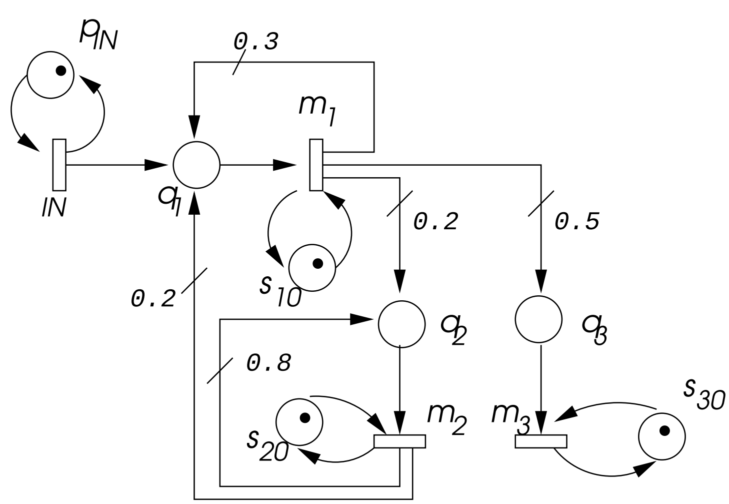


Figure 3.12: Reduced Petri net model of the queuing network in Figure 3.9.

bilities pr_{11}, pr_{12} and pr_{13} modeling the routing after the first station; t_{21}, t_{22} have associated the probabilities pr_{21}, pr_{22} modeling the routing after the second station.

The PN model includes immediate transitions and according to the discussion before, these transitions make more difficult the analysis. To reduce these transitions, let us remove the immediate transitions that are in equal conflict relationship (see Subsection 3.6). For each relationship ($\langle t_{11}, t_{12}, t_{13}\rangle$ and $\langle t_{21}, t_{22}\rangle$) one immediate transition results and removing also these transitions, the reduced PN is shown in Figure 3.11.

Using the rule presented in Figure 3.7, transitions t_1, t_2, t_3 and places s_1, s_2, s_3 can be

Table 3.1: Comparisons for the net in Fig. 3.12

	Discrete	Continuous	
		Finite server	Infinite server
Throughput(IN)	1	1	1
Throughput(m_1)	2	2	2
Throughput(m_2)	2	2	2
Throughput(m_3)	1	1	1

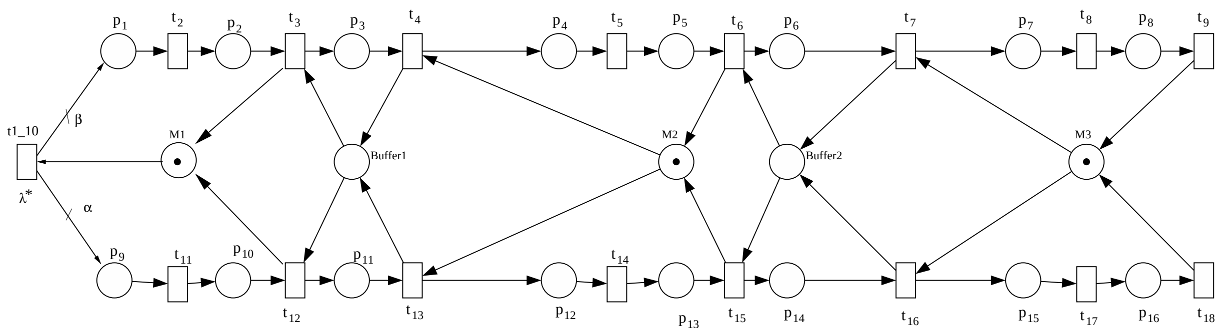


Figure 3.13: Equivalent mono-T-semiflow net of the contPN in Fig. 2.1 ($\lambda^* = \lambda_1 + \lambda_{10}, \alpha = \frac{\lambda_1}{\lambda_1+\lambda_{10}}, \beta = \frac{\lambda_{10}}{\lambda_1+\lambda_{10}}$).

eliminated and the resulted net has no immediate transitions (Figure 3.11). Remark that one place is introduced connected with a self-loop to transition IN in order to define the flow of the transitions when contPN under infinite server semantics is considered. This place is marked and will not change the evolution of the system under discrete or finite server semantics. The model obtained can be used under both continuous interpretations (finite and infinite server). For the discrete simulation, the weights of the arcs and the initial marking can be multiplied by 10 obtaining an isomorphous net.

The net system in Figure 3.12 is used in the following to compare the throughput of discrete and continuous approximations under both semantics. All configurations contain the support of a P-semiflow, because p_{IN} is the only input place in transition IN, therefore, we expect that infinite server semantics approximates better the throughput of the discrete net. Table 3.1 shows the throughput obtained by simulations and it can be observed that all are the same. In this case, the bottleneck of the system is given by the P-semiflow p_{IN}, which contains only one place, therefore the throughput under continuous infinite and finite server semantics is the same, and is equal with the throughput of the discrete system because it is mono-T-semiflow reducible with all configurations containing the support of a P-semiflow (Theorem 2.30). But what happens when the bottleneck is given by a P-semiflow containing more places?

3.7.2 Example 2

Let us consider the production system presented in Fig. 2.1 that has the following P-semiflows: $\boldsymbol{y_1} = p_1 + p_2 + M_1 + p_9 + p_{10}$ (corresponds to the state of machine M_1), $\boldsymbol{y_2} = p_3 + Buffer_1 + p_{11}$

(Buffer 1), $\boldsymbol{y_3} = p_4 + p_5 + M_2 + p_{12} + p_{13}$ (machine M_2), $\boldsymbol{y_4} = p_6 + Buffer_2 + p_{14}$ (Buffer 2) and $\boldsymbol{y_5} = p_7 + p_8 + M_3 + p_{15} + p_{16}$ (machine M_3). We have computed simulations for the continuous model under infinite and finite (single-server) server semantics using $\boldsymbol{\lambda} = \mathbf{1}$, $Buf1 = Buf2 = 10$ and we have compared it with the results obtained in the case of discrete net system (exponential distribution for transitions). This model is mono-T-semiflow reducible and can be transformed into an equivalent mono-T-semiflow net (see Fig. 3.13) without changing the loading/scheduling policy for the underlying manufacturing system. In fact, the net in Fig. 2.10 has two T-semiflows that correspond to two production lines and for the particular value of $\lambda_1 = \lambda_{10}$, the "global" T-semiflow will be $\boldsymbol{x} = \mathbf{1}$. Therefore, in steady state all the transitions will run at the same speed.

Measuring the flow of transition T_7 we have obtained the following results: $Th(T_7) = 0.186$ for the discrete model, $Th(T_7) = 0.2$ for the continuous model under infinite server semantics and $Th(T_7) = 1$ for the continuous model under finite server semantics. The results are showing clearly that continuous infinite server semantics fits here much better with the discrete results.

This net is mono-T-semiflow reducible and (as it will be seen afterwards) the equilibrium configuration contains a P-semiflow. Therefore, the throughput of the system will be given by the slowest P-semiflow. Moreover the net system is live and in that case the exact throughput of the continuous system under infinite server semantics can be computed in polynomial time (is equal with the LPP bound (2.21)) [42]. Hence, infinite server semantics is a better approximation than finite server semantics which is too optimistic in this case.

In this example, every configuration contains a P-semiflow. Let us try to find a configuration that does not contain the support of a P-semiflow. Starting with the P-semiflow corresponding to M_3 (i.e. $\boldsymbol{y_5} = p_7 + p_8 + M_3 + p_{15} + p_{16}$), the places p_7, p_8, p_{15}, p_{16} are essential covers (because their output transitions have only one input place) so should belong to the configuration. In order to not include this P-semiflow, the place M_3 should be not taken, forcing us to include p_6 and p_{14} in order to restrict the flow of t_7 and t_{16}. Now, taking into account the P-semiflow $\boldsymbol{y_4} = p_6 + Buffer_2 + p_{14}$, $Buffer_2$ cannot be taken, so p_5 and p_{13} are needed to limit the flow of t_6 and t_{15} respectively. p_4 and p_{12} are essential covers and will be taken. Observing $\boldsymbol{y_3} = p_4 + p_5 + M_2 + p_{12} + p_{13}$, M_2 cannot be taken, and so p_3 and p_{11} have to be in the configuration. Watching to $\boldsymbol{y_2} = p_3 + Buffer_1 + p_{11}$, place $Buffer_1$ will not belong to the configuration. But then, $p_2, p_1, M_1, p_{10}, p_9$ have to be in the configuration. However, all these places are the support of a P-semiflow (the one corresponding to the state of the first machine), $\boldsymbol{y_1} = p_1 + p_2 + M_1 + p_9 + p_{10}$. This means that no configuration without P-semiflow exists.

3.7.3 Example 3

In this example, we will apply some of the previous rules for the elimination of the immediate transitions to a manufacturing example from the literature and see how the simulation time is improved. This net can be seen in Figure 3.14, adding the arcs indicated in the text to avoid deadlock situations. Transitions $part_a$ and $part_b$ are in EQ relation and can be grouped, and the transition that appears can be fused with the previous one. Transitions $outM_2^a$ and $outM_2^b$ can be fused with their previous transitions (rule in Fig. 3.6). Transitions inM_1^a and inM_3^b can be reduced applying the rule shown in Fig. 3.8. Hence, the only immediate transitions that remain are $outM_1^a$, inM_2^a, inM_2^b and $outM_3^b$.

The results can be seen in Table 3.2. First, observe that throughput at steady state is quite

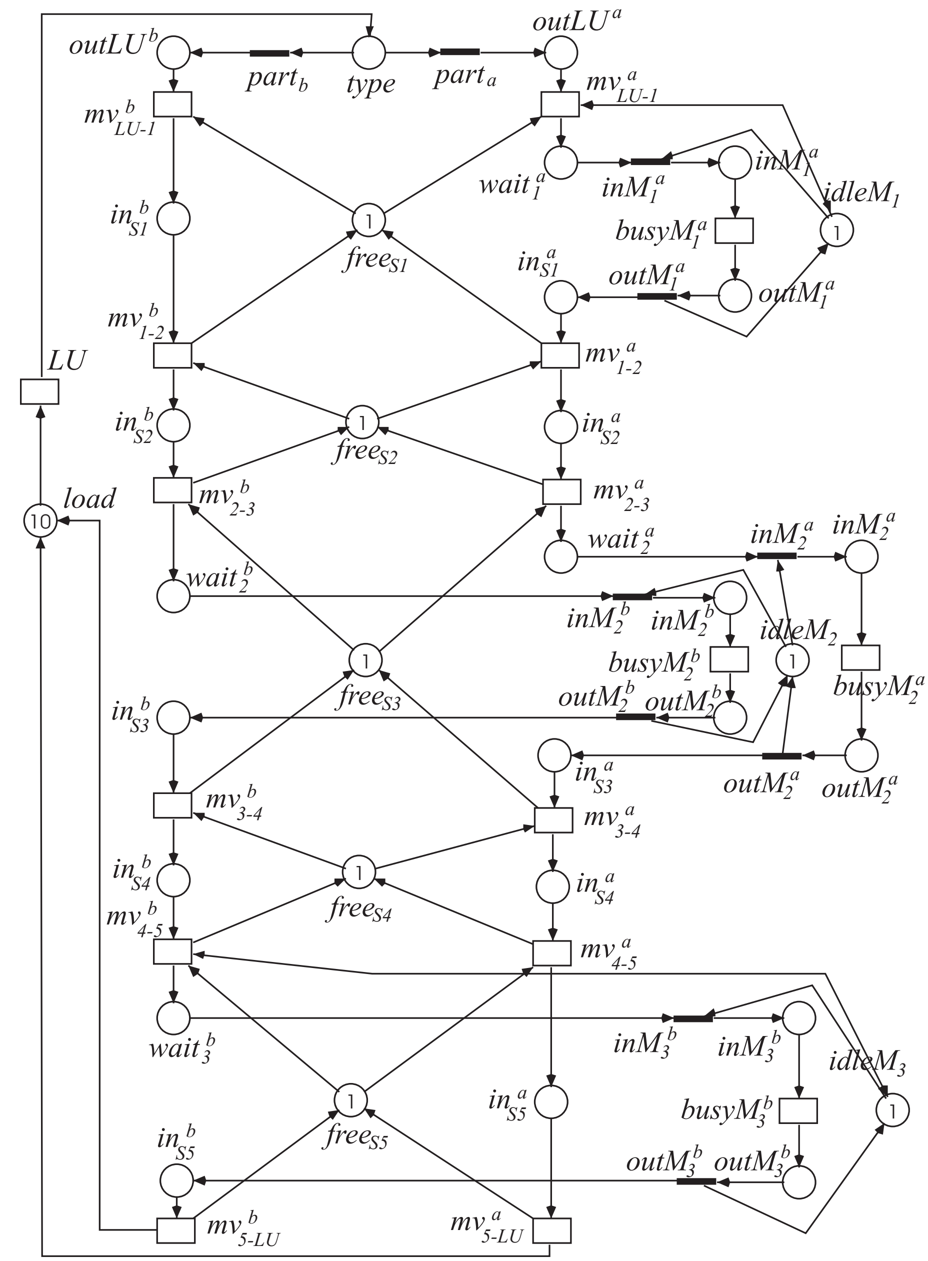


Figure 3.14: A manufacturing example taken from [1], fig. 97

Table 3.2: Comparisons for the net in Fig. 3.14

	Discrete	Continuous	
		Original	Reduced
Throughput(LU)	0.0446	0.051	0.051
Throughput(mv^a_{5-LU})	0.0357	0.040	0.040
Throughput(mv^b_{5-LU})	0.0089	0.010	0.010
Effort (in seconds)		197	146

similar to the discrete case. Since all the reduction techniques we have applied are exact, the performance of the original continuous net and of the reduced one are the same. The simulation time has been reduced approximately in 25%.

Chapter 4

Observability of continuous Petri nets with infinite server semantics

Summary

Observability of continuous Petri nets under infinite server semantics is discussed in this chapter. The first goal is to give some criteria for observability of general contPNs system that are non-linear systems, extending the results in [40] for JF class. Second, structural observability is studied trying to determine the places that should be measured such that this observability holds, a problem different from the one in [40] where the problem has been the determination of the set of places that are structurally observable from a set of measured places. Third, a new concept of observability of contPN is introduced, called generic observability using the results for structural linear systems. And, fourth, an algorithm to compute the set of places with minimum cost that ensures the observability of a JF contPN is given.

4.1 Observability: basic concepts

4.1.1 Observability of linear systems

An *unforced* (i.e., without control inputs) time invariant linear system is expressed by the following equations:

$$\begin{cases} \dot{\boldsymbol{x}}(\tau) = \boldsymbol{A} \cdot \boldsymbol{x}(\tau) \\ \boldsymbol{y}(\tau) = \boldsymbol{S} \cdot \boldsymbol{x}(\tau) \end{cases} \tag{4.1}$$

where $\boldsymbol{x}(\tau)$ is the state of the system and $\boldsymbol{y}(\tau)$ is the output, i.e., the set of measured variables. Knowing matrices $\boldsymbol{A}$ and $\boldsymbol{S}$, and being able to watch the evolution of $\boldsymbol{y}(\tau)$, a linear system is said to be *observable* if it is always possible to compute its initial state, $\boldsymbol{x}(\tau_0)$ (in fact, since the system is deterministic, knowing the state at the initial time is equivalent to knowing the state at any time).

In Systems Theory a very well-known observability criterion exists which allows to decide whether a continuous time invariant linear system is observable or not. Besides, several approaches exist to compute the initial state of a continuous time linear system that is observable.

Given an unforced linear system (4.1), the output of the system and the *observability matrix* are:

$$\boldsymbol{y}(\tau) = \boldsymbol{S} \cdot e^{\boldsymbol{A} \cdot \tau} \cdot \boldsymbol{x}(\tau_0) \tag{4.2}$$

$$\vartheta = \begin{bmatrix} \boldsymbol{S}^T, & (\boldsymbol{S}\boldsymbol{A})^T, & \cdots & ,(\boldsymbol{S}\boldsymbol{A}^{n-1})^T \end{bmatrix}^T \tag{4.3}$$

Proposition 4.1. *[46, 58] Equation (4.2) is solvable* $\forall \boldsymbol{x}(\tau_0), \forall \tau > 0$ *iff the observability matrix* ϑ *has full rank* ($rank(\vartheta) = n$).

The initial state can be obtained solving the following system of equations that has a unique solution under the rank condition:

$$\begin{bmatrix} \boldsymbol{y}(0) \\ \frac{d}{dt}\boldsymbol{y}(0) \\ \frac{d^2}{dt^2}\boldsymbol{y}(0) \\ \vdots \\ \frac{d^{n-1}}{dt^{n-1}}\boldsymbol{y}(0) \end{bmatrix} = \vartheta \cdot \boldsymbol{x}(0) \tag{4.4}$$

An interpretation of complete observability is that there is no simplification in the transfer function between the (actions on) state variables and the output [72]. Considering a single-output system, the transfer functions vector between the state variables and the output is given by:

$$\mathscr{Y}(s) = \boldsymbol{S}(s\boldsymbol{I} - \boldsymbol{A})^{-1} = \frac{1}{\Delta(s)}[q_1(s) \dots q_n(s)] \tag{4.5}$$

If $\mathscr{Y}$(s) has a cancelation (all the polynomials $q_i(s)$ and $\Delta(s)$) have a common factor) this canceled mode cannot be observed in the output y.

4.1.2 Observability of hybrid systems

While observability is well understood in classical linear system theory [46, 58], it becomes more complex in the case of hybrid systems. Observability of hybrid systems was studied in the literature in the last years [3, 4, 8, 22, 79]. For unforced systems, i.e., systems without input, the problem is to determine either the *initial state* or the *final state* from a set of observations. A system is *(initial-state) observable* if the initial state can be determined from the output function. In this chapter we consider this observation problem. Being contPN a subclass of piecewise affine systems, we provide here some of the more relevant results regarding the observability for these systems.

Definition 4.2. *[22] A continuous-time piecewise-affine hybrid system consists of*

- *A closed convex polyhedral input set* $U \subset \mathbb{R}^m$*.*
- *An observation set* $Y \subset \mathbb{R}^p$*.*
- *A finite discrete state set* Q*.*
- *A set* E *of discrete events, comprising a set* E_{in} *of input events and a set* E_{ct} *of dynamically generated events.*
- *A discrete state transition function* ρ*, which is a partial function* $Q \times E \to Q$*.*
- *For each discrete state* $q \in Q$*,*
  - *a convex polyhedral continuous state space* $X_q \subset \mathbb{R}^{n_q}$
  - *a convex polyhedral initial state set* $X_q^{init} \subset X_q$ *given by:*

$$\mathscr{J}_q(\boldsymbol{x}) = j_q + \boldsymbol{J}_q \boldsymbol{x} = 0$$

  - *an affine system* $\mathscr{A}_q$ *given by*

$$\dot{\boldsymbol{x}}(\tau) = a_q + \boldsymbol{A}_q \cdot \boldsymbol{x}(\tau) + \boldsymbol{B}_q \cdot \boldsymbol{u}(\tau)$$

  - *an affine output map* $\mathscr{Y} : X_q \times U \to Y$ *given by*

$$\boldsymbol{y}(\tau) = s_q + \boldsymbol{S}_q \cdot \boldsymbol{x}(\tau) + \boldsymbol{D}_q \cdot \boldsymbol{u}(\tau)$$

- *For each event* $e \in E$*, and each discrete state* $q \in Q$ *such that* $\rho(q,e)$ *is defined,*
  - *a closed convex polyhedral guard set* $X_{(q,e)}^{guard} \subset X_q$*, given by:*

$$\mathscr{G}_{(q,e)}(\boldsymbol{x}) = g_{(q,e)} + G_{(q,e)} \boldsymbol{x} = 0$$

  - *an affine continuous state transition function* $\mathscr{F}_{(q,e)} : X_{(q,e)}^{guard} \to X_{\rho(q,e)}$ *given by*

$$\mathscr{F}_{(q,e)}(\boldsymbol{x}) = f_{(q,e)} + F_{(q,e)} \boldsymbol{x}$$

Definition 4.3. *A trajectory of the piecewise-affine hybrid system on the time index set* $[\tau_0, \tau_1) \subset \mathbb{R}$ *with continuous input* $\boldsymbol{u} : [\tau_0, \tau_1) \to U$ *is a right-continuous function* $(q, x) : [\tau_0, \tau_1) \to X$ *such that*

1. $\boldsymbol{x}(\tau_0) \in X^{init}_{q(\tau_0)}$.

2. *If an event $e \in E$ occurs at time τ, then*

$$\boldsymbol{x}^-(\tau) \in X^{guard}_{(q^-(\tau),e)}, \quad q(\tau) = \rho(q^-(\tau), e) \quad \text{and } \boldsymbol{x}(\tau) = \mathscr{F}_{(q^-(\tau),e)}(\boldsymbol{x}^-(\tau))$$

where $q^-(\tau) = \lim_{\tau' \nearrow \tau} q(\tau')$ *and* $\boldsymbol{x}^-(\tau) = \lim_{\tau' \nearrow \tau} \boldsymbol{x}(\tau')$.

3. *If no event occurs at time τ, then $\boldsymbol{x}$ is continuous at τ, and*

$$\frac{d}{d\tau}\boldsymbol{x}(\tau) = a_{q(\tau)} + \boldsymbol{A}_{q(\tau)}\boldsymbol{x}(\tau) + \boldsymbol{B}_{q(\tau)}\boldsymbol{u}(\tau)$$

Considering the affine system with state $\boldsymbol{x} \in \mathbb{R}^n$ evolving by $\dot{\boldsymbol{x}} = \boldsymbol{a} + \boldsymbol{A}\boldsymbol{x}$, and observations $\boldsymbol{y} \in \mathbb{R}^p$ given by $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{S}\boldsymbol{x}$. Computing derivatives of $\boldsymbol{y}$, we obtain the general formula

$$\frac{d^k \boldsymbol{y}}{d\tau^k} = \boldsymbol{S}\boldsymbol{A}^{k-1}\boldsymbol{a} + \boldsymbol{S}\boldsymbol{A}^k\boldsymbol{x}, k \geq 1$$

Denoting

$$\vartheta = \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{SA} \\ \boldsymbol{SA}^2 \\ \vdots \end{bmatrix} \quad \boldsymbol{o} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{Sa} \\ \boldsymbol{SAa} \\ \vdots \end{bmatrix} \quad \mathscr{Y}(\tau) = \begin{bmatrix} \boldsymbol{y}(\tau) \\ \dot{\boldsymbol{y}}(\tau) \\ \ddot{\boldsymbol{y}}(\tau) \\ \vdots \end{bmatrix} \tag{4.6}$$

and defining the *observability map* $\bar{\vartheta}$ by $\bar{\vartheta}(\boldsymbol{x}(\tau)) = \boldsymbol{o} + \vartheta\boldsymbol{x}(\tau)$, the *observability equation* is given by:

$$\mathscr{Y}(\tau) = \bar{\vartheta}(\boldsymbol{x}(\tau)) = \boldsymbol{o} + \vartheta\boldsymbol{x}(\tau)$$

Observe that if the system is linear and not affine, i.e. $a_{q(\tau)} = 0, \forall\tau$, the observability map is exactly the observability matrix of the corresponding linear system times the states, i.e., $\bar{\vartheta}(\boldsymbol{x}(\tau)) = \vartheta\boldsymbol{x}(\tau)$.

Related to the observation time needed for the reconstruction, a hybrid system can be observable (1) in infinitesimal time, (2) in finite time T or (3) in infinite time. In this chapter we study only the observability in infinitesimal time. A very important problem in the observability of hybrid systems is the determination of the discrete state.

Definition 4.4. *Two discrete states i and j are* distinguishable *if for any $\boldsymbol{x} \in X_i$ and $\boldsymbol{x}' \in X_j$ the observations $\boldsymbol{y}(\tau)$ (for the trajectory through $\boldsymbol{x}$) and $\boldsymbol{y}'(\tau)$ (for the trajectory through $\boldsymbol{x}'$) are different on an interval* $[0, \epsilon)$.

A sufficient condition for discrete states q and q' to be distinguishable follows immediately by considering $\mathscr{Y}(\tau)$.

Proposition 4.5. *[22] If the linear equations*

$$\bar{\vartheta}_q(\boldsymbol{x}) = \bar{\vartheta}_{q'}(\boldsymbol{x}') \tag{4.7}$$

have no solution with $(x, x') \in X_q \times X_{q'}$, *then the discrete states q and q' are distinguishable.*

There exists a necessary and sufficient condition for observability in infinitesimal time of piecewise affine systems:

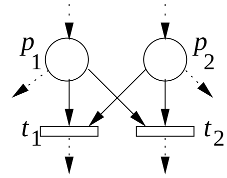


Figure 4.1: Configuration $\{(p_1, t_1), (p_2, t_2)\}$ is redundant.

Theorem 4.6. *[22] A piecewise affine hybrid system is observable in infinitesimal time iff:*

1. *All discrete states are distinguishable, and*
2. *For all discrete states, the corresponding affine system is observable.*

4.2 Observability of unforced timed continuous Petri nets

As mentioned in Section 2.3, the evolution of the timed contPNs under infinite server semantics is ruled by a set of switched linear systems. Let us consider that the marking of some places can be measured (i.e. the token charge at every time is known) due to some sensors. The other marking variables will be estimated using these measurements. Going back to (2.13), the system considered here is given by:

$$\begin{cases} \dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(\tau)) \cdot \boldsymbol{m}(\tau) \\ \boldsymbol{y}(\tau) = \boldsymbol{S} \cdot \boldsymbol{m}(\tau) \end{cases} \tag{4.8}$$

where $\boldsymbol{S}$ is a $|P_o| \times |P|$ matrix, with P_o the set of observable places, each row of $\boldsymbol{S}$ has all components zero except the one corresponding to the i^{th} measurable place that it is 1. Observe that this is a piecewise linear system since the matrix $\boldsymbol{\Pi}$ is dynamically changing with the marking but the matrix $\boldsymbol{S}$ is the same for all linear systems.

Definition 4.7. *A timed contPN system $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ under infinite server semantics is observable in infinitesimal time if it is always possible to compute its initial state $\boldsymbol{m}_0$ observing a set of $P_o \subseteq P$ places.*

In the rest of the chapter we will study observability in infinitesimal time and when we are saying that a system is observable we understand that it is observable in infinitesimal time.

Using the notions of configuration and region defined in Section 2.2.4 it may happen that for every initial marking all the markings of a region are on the border. This means that all its markings belong to other region, and so it is not necessary to consider this configuration. In the rest of the chapter we consider that these *redundant configurations* are removed.

Definition 4.8. *Let $\mathscr{C}_i$ and $\mathscr{C}_j$ be two configurations with associated regions R_i and R_j, respectively. If for all $\boldsymbol{m}_0$, $R_i \subseteq \bigcup_{j \neq i} R_j$ then $\mathscr{C}_i$ is a* redundant configuration *and the corresponding linear system is a* redundant linear system.

Example 4.9. *Let us consider the subnet in Fig. 4.1. Assume the arcs $(p_1, t_1), (p_2, t_2) \in \mathscr{C}_1$ (with associated region R_1) and the arcs $(p_1, t_1), (p_1, t_2) \in \mathscr{C}_2$ (with associated region R_2). Assume also that the other arcs of the configurations are the same, i.e., $\mathscr{C}_1 \setminus \mathscr{C}_2 = \{(p_2, t_2)\}$ and $\mathscr{C}_2 \setminus \mathscr{C}_1 = \{(p_1, t_2)\}$.*

For every initial marking $\boldsymbol{m}_0$, all reachable markings $\boldsymbol{m} \in R_1$ satisfy:

1. *$\boldsymbol{m}[p_1] \leq \boldsymbol{m}[p_2]$ since for the join t_1, the flow is given by the marking of p_1*

2. *$\boldsymbol{m}[p_2] \leq \boldsymbol{m}[p_1]$ since the flow of the join t_2 is given by the marking of p_2.*

Obviously, (1) & (2) implies $\boldsymbol{m}[p_1] = \boldsymbol{m}[p_2]$, $\forall \boldsymbol{m} \in R_1$, so R_1 is reduced to its border.

Since the flow of the other transitions is given by the same places by assumption, it is obvious that $R_1 \subset R_2$ and the linear system associated to $\mathscr{C}_2$ provides the same time-evolution for the markings $\boldsymbol{m} \in R_1$. Hence, $\mathscr{C}_1$ can be ignored and not considered as configuration that governs the evolution of the contPN system.

To see if a configuration is redundant, check if there exists a real positive marking such that the enabling degree of all the join transitions can be satisfied *only* according to the arcs in the configuration. In other words, if t is a join and (p_i, t) belongs to the configuration then check if there exists a marking $\boldsymbol{m} \in \mathbb{R}^{|P|}_{\geq 0}$ such that for all $p_j \in {}^\bullet t$, $p_j \neq p_i$, $\frac{\boldsymbol{m}[p_j]}{\boldsymbol{Pre}[p_j,t]} > \frac{\boldsymbol{m}[p_i]}{\boldsymbol{Pre}[p_i,t]}$. If such a marking does not exist, it means that the region is included into another one.

Proposition 4.10. *Let $\mathcal{N}$ be a timed contPN system and $\mathscr{C}_i$ a configuration. $\mathscr{C}_i$ is a redundant configuration iff there exists no marking $\boldsymbol{m}$ solution of:*

$$\begin{cases} \boldsymbol{m}[p_k] \in \mathbb{R}_{\geq 0} & \forall p_k \in P \\ \frac{\boldsymbol{m}[p_k]}{\boldsymbol{Pre}[p_k,t_j]} < \frac{\boldsymbol{m}[p_u]}{\boldsymbol{Pre}[p_u,t_j]}, & \forall (p_k, t_j) \in \mathscr{C}_i, \forall p_u \in \{{}^\bullet t_j\} \setminus p_k \end{cases} \tag{4.9}$$

Proof. Obviously, if (4.9) has a solution this is an interior point of the region corresponding to $\mathscr{C}_i$ and does not belong to other region, by definition.

For the reverse sense, let us assume that (4.9) has no solution. This means that for all $\boldsymbol{m}_0 \geq 0$ there exists at least one join transition t_j such that $\frac{\boldsymbol{m}[p_k]}{\boldsymbol{Pre}[p_k,t_j]} \geq \frac{\boldsymbol{m}[p_u]}{\boldsymbol{Pre}[p_u,t_j]}$ with $(p_k, t_j) \in \mathscr{C}_i$. If for all $\boldsymbol{m}$ this inequality is satisfied strictly the region is empty and can be eliminated without problems together with the corresponding configuration. Otherwise, if it is an equality, considering that the flow of t_j is given by $\boldsymbol{m}[p_u]$ not by $\boldsymbol{m}[p_k]$ it is clear that the corresponding regions include the region corresponding to $\mathscr{C}_i$. Hence, $\mathscr{C}_i$ is a redundant configuration. ☐

It may seen that if a configuration is redundant, a set of arcs has to be implicit, since they cannot define the enabling. However, it is not true, since it is not that an arc never defines the enabling, but that a combination of arcs may never define the enabling. For example, in the net in Fig. 4.1, none of the arcs is implicit, although a configuration is redundant. In this example, the redundant configuration could also have been avoided by fusing transitions t_1 and t_2 into a single one as explained in Section 3.6. However, this kind of transformation cannot always be applied, as shown in the following example.

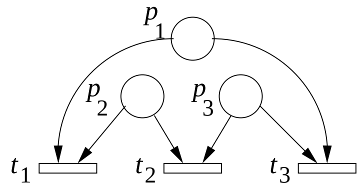


Figure 4.2: ContPN with redundant regions.

Example 4.11. *Let us consider the contPN in Fig. 4.2 and let us write the inequalities* (4.9) *corresponding to* $\mathscr{C}_1 = \{(p_2, t_1), (p_3, t_2), (p_1, t_3)\}$. *These are:*

$$\begin{cases} m_1 \in \mathbb{R}_{\geq 0} & & (a) \\ m_2 \in \mathbb{R}_{\geq 0} & & (b) \\ m_3 \in \mathbb{R}_{\geq 0} & & (c) \\ m_2 < m_1 & (p_2, t_1) \in \mathscr{C}_1, p_1 \in {}^\bullet t_1 & (d) \\ m_3 < m_2 & (p_3, t_2) \in \mathscr{C}_1, p_2 \in {}^\bullet t_2 & (e) \\ m_1 < m_3 & (p_1, t_3) \in \mathscr{C}_1, p_3 \in {}^\bullet t_3 & (f) \end{cases} \tag{4.10}$$

Combining 4.10(e) and 4.10(f) we obtain $m_1 < m_2$ *that is in contradiction with 4.10(d). Therefore,* $\mathscr{C}_1$ *is a redundant configuration.*

Synthesizing, the following assumptions are considered in this section:

(A1) The timed net structure $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is known;

(A2) The redundant configurations and regions are removed.

It is pointed out in Section 4.1.2 that for the observability of hybrid systems an important problem is the determination of the discrete state. Hence, the problem consists not only in estimating the continuous state but also the discrete one. In contPN systems, the discrete state can be deduced if the continuous state is known. However, if not every place is observed, it may happen that the observation fits with different discrete states, i.e., observing some places, it may happen that more than one system of the form (4.4) has solution. If the continuous states are on the border of some regions, it is not important which linear system is assigned, but it may happen that the solution corresponds to interior points of some regions and it is necessary to distinguish between them.

Example 4.12. *Let us consider the timed contPN in Fig. 4.3 and assume* $\boldsymbol{\lambda} = \mathbf{1}$ *and* p_3 *is the measured place. This system has two configurations:* $\mathscr{C}_1 = \{(p_1, t_1); (p_2, t_2); (p_1, t_3); (p_3, t_4)\}$ *and* $\mathscr{C}_2 = \{(p_1, t_1); (p_2, t_2); (p_2, t_3); (p_3, t_4)\}$, *corresponding to the following linear systems:*

$$\Sigma_1 = \begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \cdot \boldsymbol{m}(\tau) \\ \boldsymbol{y}(\tau) = [0, 0, 1] \cdot \boldsymbol{m}(\tau) \end{cases}$$

$$\Sigma_2 = \begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 0 \\ 0 & 1 & -1 \end{bmatrix} \cdot \boldsymbol{m}(\tau) \\ \boldsymbol{y}(\tau) = [0, 0, 1] \cdot \boldsymbol{m}(\tau) \end{cases}$$

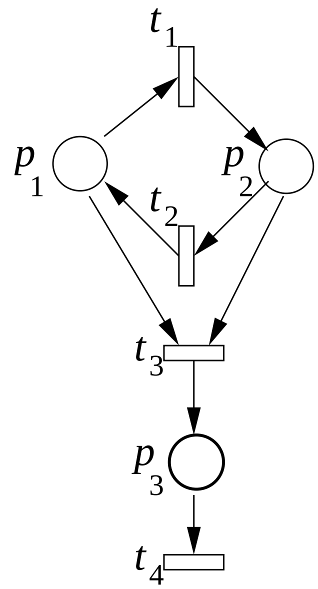


Figure 4.3: Timed contPN which has all linear systems observable measuring p_3 but which is not observable.

The observability matrices are:

$$\vartheta_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ -3 & 1 & 1 \end{bmatrix}; \quad \vartheta_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -3 & 1 \end{bmatrix}$$

which have both full rank meaning that both linear systems are observable. Let us take $\boldsymbol{m}_1 = [1,2,0]^T \in R_1 \backslash R_2$ *and* $\boldsymbol{m}_2 = [2,1,0]^T \in R_2 \backslash R_1$. *If the actual marking of the system is* $\boldsymbol{m}_1$ *then the observation is:* $\vartheta_1 \cdot \boldsymbol{m}_1 = [0,1,-1]^T$. *But, if the marking is* $\boldsymbol{m}_2$, *we observe* $\vartheta_2 \cdot \boldsymbol{m}_2 = [0,1,-1]^T$ *that it is exactly the same. Therefore, if the output of this system is* $[0,1,-1]^T$ *it is impossible to distinguish between* $\boldsymbol{m}_1$ *and* $\boldsymbol{m}_2$.

Definition 4.13. *Let* $\mathscr{C}_1$ *and* $\mathscr{C}_2$ *be two configurations with* R_1, R_2 *the associated regions.* $\mathscr{C}_1$ *and* $\mathscr{C}_2$ *are* distinguishable *if for any* $\boldsymbol{m}_1 \in R_1 \backslash R_2$ *and any* $\boldsymbol{m}_2 \in R_2 \backslash R_1$ *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 an interval* $[0,\epsilon)$.

Remark that we remove the solutions at the border $R_1 \cap R_2$ since for those points both linear systems can be used, therefore it is not important which one is chosen. This is a big difference between piecewise affine hybrid systems and contPNs: in general, for piecewise affine systems, given two states $(q,\boldsymbol{x}) \neq (q',\boldsymbol{x})$, the systems that define the evolutions are different; in contPN, if the continuous states are the same, the evolution is equal by definition using any linear system.

An immediate sufficient condition for being distinguishable, according to Prop. 4.5, is:

Proposition 4.14. *Let* $\mathscr{C}_i$, $\{i = 1,2\}$ *be a configuration,* ϑ_i *the corresponding observability matrix and* R_i *the corresponding region. If the linear equations*

$$\vartheta_1 \cdot \boldsymbol{m}_1 = \vartheta_2 \cdot \boldsymbol{m}_2 \tag{4.11}$$

have no solution $(\boldsymbol{m}_1, \boldsymbol{m}_2) \in (R_1 \setminus R_2) \times (R_2 \setminus R_1)$*, then the configurations* $\mathscr{C}_1$ *and* $\mathscr{C}_2$ *are distinguishable.*

Proof. If (4.11) has no solution then the outputs for any two markings belonging to that regions are distinct. So, given a marking in any of these regions we can determine which is the configuration that governs the evolution of the contPN system. □

Example 4.15. *Let us go back to Ex. 4.12 where the timed contPN in Fig. 4.3 is studied. Since* $\vartheta_1 \cdot \boldsymbol{m}_1 = \vartheta_2 \cdot \boldsymbol{m}_2 = [0, 1, -1]^T$*, Proposition 4.14 does not allow to conclude that* $\mathscr{C}_1$ *and* $\mathscr{C}_2$ *are distinguishable.*

(For the interpretation of this result) Let us consider the equations that govern the evolution of the system:

$$f_3 = \lambda_3 \cdot \min\{m_1, m_2\} \tag{4.12}$$

$$\dot{m}_1 = \lambda_2 \cdot m_2 - \lambda_1 \cdot m_1 - f_3 \tag{4.13}$$

$$\dot{m}_2 = \lambda_1 \cdot m_1 - \lambda_2 \cdot m_2 - f_3 \tag{4.14}$$

Summing and integrating (4.13) *and* (4.14)*, we obtain*

$$(m_1 + m_2)(\tau) = (m_1 + m_2)(0) - 2 \int_0^\tau f_3(\Theta) \cdot d\Theta \tag{4.15}$$

Obviously, if p_3 *is measured,* f_3 *can be estimated since* $f_3(\tau) = \dot{m}_3(\tau) + \lambda_4 \cdot \dot{m}_3(\tau)$*. Therefore, according to* (4.12)*, the minimum between* m_1 *and* m_2 *is estimated and according to* (4.15) *their sum is also known. These two equations are not enough to compute the markings, i.e., we have the values but it is impossible to distinguish which one corresponds to which place.*

Using the notion of distinguishable configurations, an immediate criterium for observability in infinitesimal time is:

Theorem 4.16. *A timed continuous Petri net system* $\langle \mathscr{N}, \boldsymbol{\lambda} \rangle$ *under infinite server semantics is observable in infinitesimal time iff:*

1. *All configurations are distinguishable,*
2. *For each configuration, the associated linear system is observable.*

Proof. Assume that given an observation $\bar{\boldsymbol{m}}$, there are two different markings $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ coherent with $\bar{\boldsymbol{m}}$. Since the linear systems are observable, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ belong to different configurations. But the configurations are all distinguishable, contradiction.

If the contPN is observable, for any initial marking in any configuration it must be possible to reconstruct it from observation, hence all the linear systems associated to the configurations have to be observable. Moreover, the configurations have to be distinguishable, since otherwise it would be possible to have two different markings that fit with the observation. □

4.3 Structural observability

Observability has been defined for a timed contPN system $\langle \mathscr{N}, \boldsymbol{\lambda} \rangle$, so the firing rates of the transitions are fixed. Since the firing rate represents the speed of a machine or a server, in

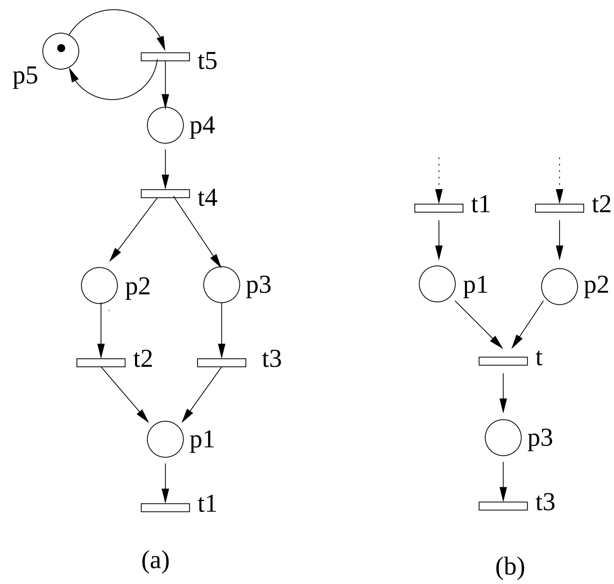


Figure 4.4: (a) A JF net that if $\lambda_2 = \lambda_3$ then m_2 and m_3 cannot be estimated from the observation of p_1; (b) A PN observable for any initial marking only if all places are measured.

many cases, an interesting problem is to study the observability for any value of its rate. Imagine that we want to design an observer and we know that in the future some machines will be replaced but we don't know exactly which one will be bought, hence their speed is not fixed. In this chapter, we are concentrate on the study of the observability of contPNs under infinite server semantics in infinitesimal time for any value of firing rate $\boldsymbol{\lambda}$. We call this problem: *structural observability*. The following assumptions are done:

(A1) The net structure $\mathcal{N}$ is known and $\boldsymbol{\lambda}$ is a parameter;

(A2) The redundant configurations and regions are removed.

Definition 4.17. *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a contPN system and $P_o \subseteq P$ the set of measured places.*

- *A place $p \in P$ is structurally observable from P_o if $\forall \boldsymbol{\lambda} > 0$, $m_0[p] = m(\tau_0)[p]$ can be computed in $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ by measuring the marking evolution of the places in P_o.*
- *Let $K(P_o)$ be the set of places structurally observable from P_o. $\mathcal{N}$ is structurally observable from P_o if every place $p \in P$ is structurally observable, i.e., $K(P_o) = P$.*

Since structural observability means that the contPN system is observable for all values of $\boldsymbol{\lambda} > 0$, we have:

Remark 4.18. *Let $\mathcal{N}$ be a contPN. If $\mathcal{N}$ is structurally observable then for all $\boldsymbol{\lambda} > 0$ $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is observable.*

Due to the graphical representation of PNs, the observation procedure has a quite interesting interpretation, going backward on the net:

Example 4.19. *Let us consider the contPN system in Fig. 4.4(a) and assume that p_2 is measured. So, the marking in p_2 is known at every time instant. Then the derivative of the marking can be estimated, and also the flow of the transition t_2 because $f_2 = \lambda_2 \cdot m_2$. Evidently, the flow of t_4 is deduced immediately using that $f_4 = \dot{m}_2 + f_2$ and then the marking of p_4 can be computed because, on the other hand, $f_4 = \lambda_4 \cdot m_4$.*

The backward procedure explained in the previous example assumes that all places on the backward path are AF (a unique input flow per place) and all transitions are JF (the min operator will stop the reasoning).

Definition 4.20. *Let $\mathcal{N}$ be a contPN. $\langle p_1, t_1, p_2, t_2, \ldots, t_k, p_{k+1}\rangle$ is a* JF & AF (JA-F) path *from p_1 to p_{k+1} if:*

- $p_1, p_2, \ldots, p_{k+1} \in P$ *and* $t_1, t_2, \ldots, t_k \in T$;
- ${}^\bullet t_i = \{p_i\}$ *and* $p_{i+1} \subseteq t_i{}^\bullet$ *with* $i = 1, \ldots k$ *(t_i is not a join transitions and has as an output place p_{i+1});*
- ${}^\bullet p_i = \{t_{i-1}\}, i = 2, \ldots, k+1$ *and* $t_i \in p_i{}^\bullet, i = 1, \ldots, k$ *(p_i is not an attribution and has as an output transition t_i).*

Definition 4.21. *Let $\mathcal{N}$ be a contPN.*

- *A place p' is* output connected *if there exists a path from p' to a measured place p.*
- *$\mathcal{N}$ is output connected if all places are output connected.*

For JA-F PN, i.e., only forks and choices may exist, structural observability is solved without difficulty using the basic backward strategy presented above (see [40] for more details).

Proposition 4.22. *[40] Let $\mathcal{N}$ be a JA-F contPN. A place p' is structurally observable if it is output connected.*

Therefore, for JA-F nets, the set P_o of places that ensures the output-connectedness of each place is computed immediately using the strongly connected components with respect to the places. In these components, transitions are not important so we can have a source transition entering in a place or an output transition from a place.

Definition 4.23. *Let $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ be a net and $\mathcal{N}' = \langle F, T', \boldsymbol{Pre}', \boldsymbol{Post}'\rangle$ a subnet of $\mathcal{N}$, i.e., $F \subseteq P$, $T' \subseteq T$ and $\boldsymbol{Pre}', \boldsymbol{Post}'$ are the restrictions of $\boldsymbol{Pre}, \boldsymbol{Post}$ to F and T'. $\mathcal{N}'$ is called a strongly connected component of $\mathcal{N}$ w.r.t. the places if for all $p_1, p_2 \in F$ there is a path from p_1 to p_2 of the form $\langle p_1, t_1, p_i, t_i, \ldots, t_j, p_j, t_2, p_2\rangle$ with $t_1 \in p_1{}^\bullet$, $p_i \in t_1{}^\bullet, \ldots, p_j \in t_j{}^\bullet$, $t_2 \in p_j{}^\bullet$, $p_2 \in t_2{}^\bullet$.*

Abusing of notation it will be said that a set of places F is a strongly connected component of $\mathcal{N}$ if $\mathcal{N}'$ is a strongly connected component of $\mathcal{N}$ with F its set of places and $T' = {}^\bullet F \cup F^\bullet$ its set of transitions.

The net in Fig. 4.5 has only one strongly connected component $F = \{p_1, p_2, p_3, p_4\}$ because a path exists connecting any two places. For example from p_1 to p_4 there is: $\langle p_1, t_1, p_3, t_2, p_2, t_4, p_4\rangle$. The net in Fig. 4.4(a) has 5 strongly connected components, each one corresponding to a place, i.e., $F_i = \{p_i\}$, $i = 1 \ldots 5$.

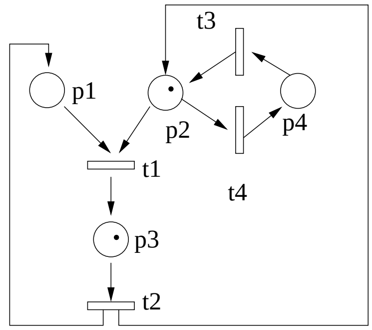


Figure 4.5: A simple contPN system.

Output connectedness is required for structural observability but also for observability. Obviously, for those places for which there is no path to an output their marking cannot be estimated because they do not affect the observed outputs. Therefore, a special interest represent the *terminal strongly connected components* because all places are connected to those components.

Definition 4.24. *A strongly connected component $\mathcal{N}' = \langle F, T', \boldsymbol{Pre}', \boldsymbol{Post}' \rangle$ of a net $\mathcal{N}$ is said to be* terminal *if there is no path from a place belonging to F to a place not in F.*

Strongly-connected components of a PN can be computed immediately, adapting the algorithm in [24] to a bipartite graph. The net in Fig. 4.5 has one strongly connected component which is obviously terminal, while the net in Fig. 4.4.(a) has only one terminal strongly connected component $F_1 = \{p_1\}$.

Proposition 4.25. *Let $\mathcal{N}$ be a JA-F contPN. $\mathcal{N}$ is (structurally) observable iff at least one place from each terminal strongly connected component is measured.*

Proof. If $\mathcal{N}$ is (structurally) observable then every place that is not measured should be output connected to a place that is measured. Therefore at least one place from each terminal strongly connected component should be measured.

On the contrary, if at least one place from each terminal strongly connected component is measured, every place is output connected and the net is structurally observable according to Prop. 4.22. If $\mathcal{N}$ is structurally observable, according to Remark 4.18 it is observable for a particular $\boldsymbol{\lambda}$. □

Therefore, the minimum number of places to ensure the (structural) observability of a JA-F contPN is equal to the number of terminal strongly connected components. Let us see what happens when joins ($|^\bullet t| \geq 2$) occur. According to (2.9), this introduces nonlinearity into the flow definition due to the minimum function and this will cause problems in the observation procedure.

Example 4.26. *Let us consider the contPN subsystem in Fig. 4.4(b). It is (structurally) observable iff places $P_o = \{p_1, p_2, p_3\}$ are measured.*

Place p_3 must be measured (it is a terminal strongly-connected component). Using the marking of p_3, the flow of transition t_3 can be computed as $f_3 = \lambda_3 \cdot m_3$. The derivative of the marking of p_3 and the flow of transition t_3 permit to compute the flow of transition t using

$\dot{m}_3 = f_t - f_3 \Longrightarrow f_t = \dot{m}_3 + f_3$. On the other hand, this flow is equal to $f_t = \lambda_t \cdot \min\{m_1, m_2\}$. In the last expression, f_t and λ_t are known which implies that the minimum of m_1 and m_2 can be evaluated. If $m_1 \leq m_2$ (place p_2 does not constraint the firing of t), m_1 equals to the minimum and p_2 must be measured. Identically, if at a certain moment $m_2 \leq m_1$, p_1 should be measured. Therefore, if no information regarding how m_1 and m_2 compare is known, then the only solution *for observability is to measure both p_1 and p_2. Moreover, since p_1, p_2 and p_3 are measured, the observability space of this system is the same as that of the system obtained removing the join transition t.*

Proposition 4.27. *[52] Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a timed AF contPN and $\mathcal{N}'$ obtained from $\mathcal{N}$ by just removing all join transitions together with its input and output arcs. $\mathcal{N}$ is (structurally) observable iff $\mathcal{N}'$ is (structurally) observable.*

Proof. Let us see that if $\mathcal{N}'$ is (structurally) observable then $\mathcal{N}$ is (structurally) observable. In $\mathcal{N}'$ all places must be output connected to measured places. Then, adding the join transitions in $\mathcal{N}$ the net system is also output connected and the same JA-F paths exist. Hence the observation procedure is the same, going backwards on the paths.

For the reverse, if $\mathcal{N}$ is structurally observable it is observable for any value of $\boldsymbol{\lambda}$. $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is observable iff every linear system is observable and the configurations are distinguishable (Prop. 4.16). Let us assume, for simplicity, that $\mathcal{N}$ has only one join t with $p_1, p_2 \in {}^\bullet t$ (see Fig. 4.4(b)) (the proof can be easily extended).

The system has two configurations $\mathscr{C}_1$ and $\mathscr{C}_2$ and let $(p_1, t) \in \mathscr{C}_1$ and $(p_2, t) \in \mathscr{C}_2$, respectively. The linear system associated to $\mathscr{C}_1$ is observable, hence from p_2 a path should exist to one output, but not using the arc (p_2, t) because the marking of p_2 is not giving the enabling degree of t in this configuration (the arc (p_1, t) can be used). Let us denote by $\mathscr{P}_2$ the path from p_2 to one output. Analogously, the linear system associated to $\mathscr{C}_2$ is observable, then a path from p_1 to an output should exit but not using the arc (p_1, t). Let $\mathscr{P}_1$ be the path from p_1 to one output.

If $\mathscr{P}_1$ and $\mathscr{P}_2$ do not contain the transition t, it is obvious that the both nets are observable (with and without t) since p_1 and p_2 are output connected to the same outputs in both systems (linear and nonlinear). If only one of the paths contains t, for example $\mathscr{P}_1$, it has the form: $\mathscr{P}_1 = \langle p_1, \cdots, p_2, t, \cdots, p_i \rangle$, with p_i a measured place, since the arc (p_1, t) cannot be considered. But it is assumed there exists a path from p_2 to an output not containing t, therefore p_1 is also connected to other output not containing t and the same as before, both systems are observable.

If $\mathscr{P}_1$ and $\mathscr{P}_2$ both contain t, and there are no other outputs to which they are connected, then the configurations are not distinguishable. Indeed, the same evolution is obtained taking $\frac{\boldsymbol{m}[p_1]}{\boldsymbol{Pre}[p_1,t]} = q_1$ and $\frac{\boldsymbol{m}[p_2]}{\boldsymbol{Pre}[p_2,t]} = q_2$, $q_1 \neq q_2$, at the output through t, or $\frac{\boldsymbol{m}[p_1]}{\boldsymbol{Pre}[p_1,t]} = q_2$ and $\frac{\boldsymbol{m}[p_2]}{\boldsymbol{Pre}[p_2,t]} = q_1$ (see Example 4.15). □

The previous theorem does not hold when the net is not AF.

Example 4.28. *Let us consider the contPN system in Fig. 4.6 with $\boldsymbol{\lambda} = [a, 1, 2, 3, 4]^T$, $a \in \mathbb{R}_{\geq 0}$ and p_5 measured. This net is not AF and has a join in t_1. Notice that the linear system obtained removing the join t_1 is observable. For the non-linear system, the corresponding linear systems are:*

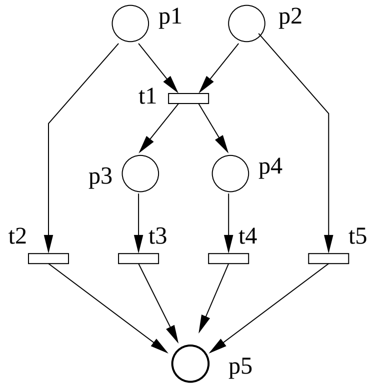


Figure 4.6: ContPN system used in Ex. 4.28.

$$\Sigma_1 = \begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -1-a & 0 & 0 & 0 & 0 \\ -a & -4 & 0 & 0 & 0 \\ a & 0 & -2 & 0 & 0 \\ a & 0 & 0 & -3 & 0 \\ 1 & 2 & 3 & 4 & 0 \end{bmatrix} \cdot \boldsymbol{m}(\tau) \\ \boldsymbol{y}(\tau) = [0,0,0,0,1] \cdot \boldsymbol{m}(\tau) \end{cases}$$

$$\Sigma_2 = \begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -1 & -a & 0 & 0 & 0 \\ 0 & -4-a & 0 & 0 & 0 \\ 0 & a & -2 & 0 & 0 \\ 0 & a & 0 & -3 & 0 \\ 1 & 2 & 3 & 4 & 0 \end{bmatrix} \cdot \boldsymbol{m}(\tau) \\ \boldsymbol{y}(\tau) = [0,0,0,0,1] \cdot \boldsymbol{m}(\tau) \end{cases}$$

with observability matrices:

$$\vartheta_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 4\cdot a-1 & -8 & -6 & -12 & 0 \\ -(4\cdot a-1)\cdot(a+1)-10\cdot a & 12 & 36 & 0 & \\ ((4\cdot a-1)\cdot(a+1)+10\cdot a)\cdot(a+1)+16\cdot a & -128 & -24 & -108 & 0 \end{bmatrix}$$

$$\vartheta_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ -1 & 4\cdot a-8 & -6 & -12 & 0 \\ 1 & -(4\cdot a-8)\cdot(a+4)-17\cdot a & 12 & 36 & 0 \\ -1 & ((4\cdot a-8)\cdot(a+4)+17\cdot a)\cdot(a+4)+47\cdot a & -24 & -108 & 0 \end{bmatrix}$$

Computing the determinants of the observability matrices, we have: $det(\vartheta_1) = 192\cdot a^3 - 912\cdot a^2 + 720\cdot a + 288$, *which has two positive real roots, and* $\det(\vartheta_2) = -96\cdot a^3 - 408\cdot a^2 - 216\cdot a + 288$, *with one positive real root. Obviously, if* λ_1 *is equal to one of these roots, the contPN system will not be observable since one of the corresponding linear systems will not be observable.*

Hence, for some particular values of $\boldsymbol{\lambda}$, the system obtained removing the join is observable but the original system (with join) is not observable.

Hence, for AF nets, the observability can be studied using the linear system theory since their observability space with respect to the JF net, obtained removing the joins, includes only some additional information (minimum of some weighted markings) that cannot provide information to estimate markings. For this class, the set P_o of places that ensures structural observability of the system is computed as in JA-F case using Prop. 4.25, after the joins are eliminated. Notice that due to the elimination of the joins, several unconnected PN can be obtained. In this case, Prop. 4.25 is applied for each connected component.

Proposition 4.29. *The structural observability of an AF contPN can be solved in polynomial time at the graph level.*

In the case of CEQ nets, observability can be studied also using linear system theory since the joins can be eliminated as in the case of AF nets.

Proposition 4.30. *Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a timed CEQ contPN system and $\mathcal{N}'$ obtained from $\mathcal{N}$ by just removing all join transitions together with its input and output arcs. $\mathcal{N}$ is (structurally) observable iff $\mathcal{N}'$ is (structurally) observable.*

Proof. First, notice that the net $\mathcal{N}$ can be transformed into an equivalent one obtained through fusion of the CEQ (see Section 3.6). Hence, it can be assumed that the net has no choice, i.e., for every join transition t, for all $p \in {}^\bullet t$, $p^\bullet = \{t\}$.

"$\Longrightarrow$" Let t be a join, and $p, p' \in {}^\bullet t$. For those markings in the region defined by the configuration that contain (p, t), the only way to observe p' for any $\boldsymbol{\lambda}[t]$ is to measure it, since its only output is t and that transition cannot be used to estimate it.

The same can be said for p, hence both places have to be measured and removing their output transition cannot affect the (structural) observability of the system.

"$\Longleftarrow$"If $\mathcal{N}'$ is structurally observable it is observable for a given $\boldsymbol{\lambda}$. Since $\mathcal{N}'$ is obtained from $\mathcal{N}$ (that it is CEQ by assumption) and the joins are removed, all input places in the joins of $\mathcal{N}$ have no output transition in $\mathcal{N}'$. But $\mathcal{N}'$ is observable, hence all these places should be measured because cannot be estimated with others measurements. Measuring these places, the linear systems of $\mathcal{N}$ are distinguishable. Moreover, all linear systems are observable since the observability does not depends on the firing rates of the output transitions of the measured places (Prop. 6 in [40]). According to Theorem 4.16 $\mathcal{N}$ is observable. □

Unfortunately, the elimination of joins cannot be performed in general, because the observability of nets with attributions should be study globally, not locally (see Ex. 4.28). For CEQ nets, that in principle have attributions, all joins can be removed (Prop. 4.30), but only because their input places should be measured which is not true in general. This can be seen in Ex. 4.28 taking $\boldsymbol{\lambda} = [a, 1, 2, 3, 4]^T$ with a different from the roots of $det(\vartheta_1)$ and $det(\vartheta_2)$. Since t_1, t_2, t_5 are not in CEQ relation, it is not sure that all their input places have to be measured. Indeed, this contPN system is observable measuring only p_5.

In summary, according to Prop. 4.27, given an AF contPN, joins can be removed without affecting the observability. This holds also for CEQ nets (Prop. 4.30). In the new net all the conflicts are CEQ without synchronizations (if $t_1, t_2 \in p^\bullet, {}^\bullet t_1 = {}^\bullet t_2 = p$) and so, the net can be mapped into CF [42]. Therefore, *forks* and *choices* do not pose any problem for observability (JA-F case), while *joins* are real "barriers" in the backward procedure. Let us now consider attributions, the only local construction of nets not yet studied. This construction can introduce zeros in the transfer functions, possibly leading to pole-zero cancelations, thus loss of observability.

Example 4.31. *Let us consider the JF contPN system in Fig. 4.4(a) (it has an attribution in p_1). Assume that p_1 is measured. This system is a continuous (time-invariant) linear system. If we consider that the input of the system is the input flow to p_4 and the measured output is m_1, the equivalent linear system $\dot{\boldsymbol{x}}(\tau) = \boldsymbol{A} \cdot \boldsymbol{x}(\tau), \boldsymbol{y}(\tau) = \boldsymbol{S} \cdot \boldsymbol{x}(\tau)$ has:*

$$\boldsymbol{A} = \begin{pmatrix} -\lambda_1 & \lambda_2 & \lambda_3 & 0 \\ 0 & -\lambda_2 & 0 & \lambda_4 \\ 0 & 0 & -\lambda_3 & \lambda_4 \\ 0 & 0 & 0 & -\lambda_4 \end{pmatrix},$$

$$\boldsymbol{S} = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

The transfer function vector between the input flow in places and the output, using Equation (4.5) is:

$$\mathcal{Y}(s) = \frac{1}{(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)} H^T \tag{4.16}$$

where:

$$H = \begin{bmatrix} (s+\lambda_2)\cdot(s+\lambda_3)\cdot(s+\lambda_4) \\ \lambda_2\cdot(s+\lambda_3)\cdot(s+\lambda_4) \\ \lambda_3\cdot(s+\lambda_2)\cdot(s+\lambda_4) \\ (\lambda_2\cdot(s+\lambda_3)+\lambda_3\cdot(s+\lambda_2)) \end{bmatrix} \tag{4.17}$$

In Equations (4.16) *and* (4.17)*, if $\lambda_2 = \lambda_3$ there is a pole-zero simplification in* all *elements of vector $\mathcal{Y}(s)$ leading to the conclusion that the system is not observable [72]. If $\lambda_2 \neq \lambda_3$, but $\lambda_4 = \frac{2\cdot\lambda_2\cdot\lambda_3}{\lambda_2+\lambda_3}$, there is another simplification and the system is not observable. Consequently, when an attribution appears, particular values of $\boldsymbol{\lambda}$ exist such that the observability is lost. Moreover, it is not a local property, but depends on the whole net structure.*

Usually, if p is an attribution place and $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$, with $t_1, t_2 \in {}^\bullet p$ then there exists a pole-zero cancelation and an additional place should be measured. But this is not a general rule as illustrated in the following example.

Example 4.32. *Let us consider the net in Fig. 4.7 with $\boldsymbol{\lambda} = \mathbf{1}$ and assume that p_2 is measured. Then p_4 and p_5 cannot be estimated directly, but a linear combination of the markings of these places is known (place p_{45} in Fig. 4.7). Going backwards, p_1 is estimated and, even although p_1 is an attribution, since p_2 is measured p_3 is also estimated. Using the marking of p_3, p_4 is estimated and through the linear combination of p_{45}, p_5 as well. Therefore, the system is observable measuring p_2 for any values of firing rates of the transitions. As said before: observability is a global property of the system.*

4.4 Generic observability

In Ex. 4.31, the pole-zero cancelation due to the attribution happens for very specific values of $\boldsymbol{\lambda}$. If the firing rates of the transitions are chosen randomly in $\mathbb{R}^+$, the probability to ob-

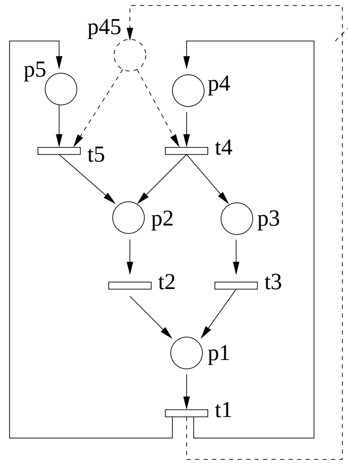


Figure 4.7: A JF net that is observable measuring the attribution place p_2 even if $\lambda_4 = \lambda_5$.

tain this cancelation is null. Hence, a concept weaker than structural observability can be studied. It is defined following the ideas presented in [30, 23] for linear systems hence we consider JF nets for which the behavior is linear and not piecewise linear as in general case. Regarding our assumptions, we consider:

(A1) The net structure $\mathcal{N}$ is known and $\boldsymbol{\lambda}$ is a parameter;

(A2) $\mathcal{N}$ is JF.

According to the results in the previous section generic observability can be studied also for AF and CEQ nets. In these cases, as explained before, joins can be removed and the obtained JF net is observable iff the original net is observable. In a JF net, choices are CEQ, thus can be transformed into forks (see Section 3.6), and a JC-F net is obtained. Therefore, we can assume that the nets are JC-F.

Definition 4.33. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a JF contPN system and P_o a set of measured places. $\mathcal{N}$ is* weakly structural *or* generically observable *from P_o if $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is observable for all values of $\boldsymbol{\lambda}$ outside a proper algebraic variety of the parameter space.*

Connection between structural and generic observability is obvious. If $\mathcal{N}$ is structurally observable then it is generically observable. The reverse is not true (see Example 4.31). Connections between observability for a given $\boldsymbol{\lambda}$ and generic observability are also obtained immediately.

Proposition 4.34. *Let $\mathcal{N}$ be a contPN, $\boldsymbol{\lambda}$ a set of rates. Generic observability for $\mathcal{N}$ does not imply observability for a particular $\boldsymbol{\lambda}$.*

In [23], generic observability is studied for structured linear systems using an *associated graph*; observability is guaranteed when there exists a state-output connection for every state variable (the system is said to be *output connected*) and no *contraction* (defined after) exists.

The associated graph of an unforced linear system (4.1), $G = (Z, W)$ is defined by a vertex set Z and an edge set W [23]. The vertex set $Z = X \cup Y$ with X the set of state vertices and

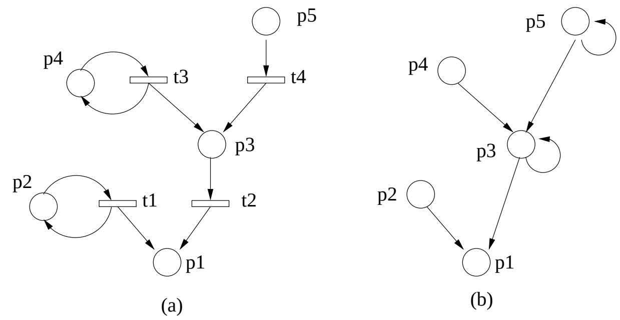


Figure 4.8: (a) A JF ContPN; (b) Associated graph.

Y the set of output vertices. Denoting (v, v') for a direct edge from the vertex $v \in Z$ to a vertex $v' \in Z$, the edge set W is described by $W_A \cup W_S$ with $W_A = \{(x_j, x_i) | A[i, j] \neq 0\}$ and $W_S = \{(x_j, y_i) | S[i, j] \neq 0\}$.

The transformation of a JF net into its corresponding *associated directed graph* can be computed as follows (see Fig. 4.8). The vertex set Z is given by the set P of places (i.e. $Z = P$). The edge set W is computed as: $W = \{(p_i, p_j) | p_j \in (p_i{}^\bullet)^\bullet \bigwedge p_i \neq p_j\} \cup \{(p_i, p_i) | \exists t \in p_i{}^\bullet, \boldsymbol{Pre}[p_i, t] \neq \boldsymbol{Post}[p_i, t]\}$. The first set adds an edge from a place p_i to all places $(p_i{}^\bullet)^\bullet$ since the dynamic matrix has a non null entry and prevents adding an edge in the case of a self-loop. The second subset will add a self-loop in the associated graph for any place with $\boldsymbol{Pre}[p_i, t] \neq \boldsymbol{Post}[p_i, t]$, i.e., the marking of p_i will change firing t, implying that the dynamical matrix has an non zero entry.

Definition 4.35. *Let $\mathcal{N}$ be a contPN system and $G(\mathcal{N})$ its associated graph with vertex set Z and edge set W. Consider a set S made of k_S state vertices. Denote $E(S)$ the set of vertices w_i for $i = 1, \cdots, l_S$ of Z, such that there exists an edge $(x_j, w_i) \in W$ with $x_j \in S$. S is said to be a* contraction *if $k_S - l_S > 0$.*

Based on the procedure to generate the associated graph (Fig. 4.8), and using Prop. 1 in [23], the following is true:

Proposition 4.36. *Let $\mathcal{N}$ be a contPN and $G(\mathcal{N})$ its associated graph. $\mathcal{N}$ is generically observable iff:*

1. *$\mathcal{N}$ is output connected*

2. *$G(\mathcal{N})$ contains no contraction.*

Example 4.37. *Let us consider the contPN in Fig. 4.8(a) whose associated graph is sketched in Fig. 4.8(b). Taking $S = \{p_2, p_3, p_4, p_5\}$ ($k_S = 4$), $E(S) = \{p_1, p_3, p_5\}$ ($l_S = 3$). Thus, the net has a contraction ($k_S - l_S = 4 - 3 = 1$), so it is not generically observable. This happens because the flows of the transitions t_1 and t_3 are constant and measuring p_1 it is impossible to distinguish between these two constant incoming flows.*

In the case of pure contPN systems, the necessary and sufficient condition of generic observability can be simplified. Since the associated graph of a pure PN has in every node a self-loop (under infinite server semantics, if p_i has at least one output transition t_j the derivative of the marking is: $\dot{m}_i = \cdots - \lambda_j \cdot m_i + \cdots$). Therefore, no contraction can exist and the only remaining condition in Prop. 4.36 is the output connectedness.

Corollary 4.38. *Let $\mathcal{N}$ be a pure JF contPN. $\mathcal{N}$ is generically observable iff at least one place from each terminal strongly connected component is measured.*

4.5 Minimum cost observability of JF nets

4.5.1 Problem statement

In this section, minimum cost observability of JF nets is studied. We have:

(A1) The timed net structure $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is known;

(A2) $\mathcal{N}$ is JF net.

As done for generic observability, the problem of minimum cost can be studied also for AF and CEQ nets since in these cases joins can be removed without affecting the observability (see Section 4.3). Moreover, JF nets can be transformed in JC-F and this class is considered from now.

Definition 4.39. *Let $w(p) > 0$ be the cost to measure place p. The observability cost for a given set P_o is $w(P_o) = \sum_{p_i \in P_o} w(p_i)$; the* minimal cost observability *problem is to determine a set P_o with minimum cost that makes $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ observable for every initial marking $\boldsymbol{m}_0$.*

Minimal cost observability problem can be seen as a Set Covering Problem (SCP), which is NP-hard in the strong sense [31]. For each set of places P_o, let K_{P_o} be the set of observable places. It will be said that P_o is covering K_{P_o}. The problem is to determine a set P_{o_i} with minimum cost such that the covered elements K_i contain all the places of the net. Unfortunately, the number of covering elements (P_{o_i}) is not limited to the isolated places, but also subsets of places have to be considered.

Example 4.40. *Let us consider the contPN system in Fig. 4.4(a) without place p_5 and transition t_5, with $\lambda_2 = \lambda_3$ and $\lambda_4 \neq \lambda_2$. The set of observable places from $P_{o_1} = \{p_1\}$ is $K_1 = \{p_1, p_4\}$ (m_1 being measured, $m_2 + m_3$ can be estimated and also m_4). From $P_{o_2} = \{p_2\}$ the set of observable places is $K_2 = \{p_2, p_4\}$; from $P_{o_3} = \{p_3\}$ we obtain $K_3 = \{p_3, p_4\}$; and finally from $P_{o_4} = \{p_4\}$, $K_4 = \{p_4\}$. According to this, a solution is to measure p_1, p_2 and p_3 because that way all the places are covered. Nevertheless, that solution is not in general an optimal one.*

If $\lambda_2 = \lambda_3$, the incoming flows into the attribution in p_1 (f_2 and f_3) are indistinguishable: $\dot{m}_1 = \lambda_2(m_2 + m_3) - \lambda_1 m_1$. But, if the flow through t_2 is known, the flow through t_3 is known too, and viceversa.

Indeed, measuring $P_{o_5} = P_{o_1} \cup P_{o_2} = \{p_1, p_2\}$ or $P_{o_6} = P_{o_1} \cup P_{o_3} = \{p_1, p_3\}$ the system is observable, i.e., $K_5 = K_6 = \{p_1, p_2, p_3, p_4\}$. It is interesting to notice that $K_5 \supset K_1 \cup K_2$ and $K_6 \supset K_1 \cup K_3$. The optimal solution for the observability is: $w(p_1) + \min\{w(p_2), w(p_3)\}$.

Remark 4.41. *Minimal cost solutions need not to be of minimal cardinality.*

This can be seen considering the net system in Fig. 4.9, assuming $\lambda_3 = \lambda_4$, $\lambda_5 = \lambda_6$ and the other λs different. Measuring $P_{o_1} = \{p_4\}$ or $P_{o_2} = \{p_3, p_5\}$ the system is observable. If $w(P_{o_2}) < w(P_{o_1})$ then the minimal solution is P_{o_2}, even if $|P_{o_2}| > |P_{o_1}|$.

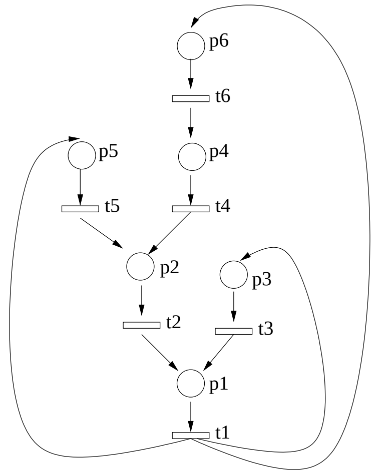


Figure 4.9: If $\lambda_3 = \lambda_4$ and $\lambda_5 = \lambda_6$ the JF net is not observable measuring p_3 or p_5 but observable measuring both.

4.5.2 Brute force method

Considering that n is the number of places, the brute force approach to solve this problem is to try all subsets of places of size n, $n-1$, $\cdots$, 1. Obviously, two monotonicity properties can be used to reduce the complexity: (1) if the system is not observable measuring P_{o_i} then it is not observable for any P_{o_j} with $P_{o_j} \subset P_{o_i}$, and (2) if the system is observable measuring P_{o_i} or P_{o_j} and $P_{o_j} \subset P_{o_i}$ then P_{o_i} cannot be the optimal solution because its cost is greater than the one of P_{o_j}.

Example 4.42. *Let us consider the net in Fig. 4.9 with* $\boldsymbol{\lambda} = \mathbf{1}$ *and* $w = (1,1,2,5,3,4)$. *A brute force method to solve minimum cost observability consist in:*

- *Considering subsets of length* 6*:*
  - $P_{o_1} = \{p_1, p_2, p_3, p_4, p_5, p_6\}$ *ensures the system observability at cost* $w(P_{o_1}) = 16$ *therefore it is kept as a provisional solution*
- *With subsets of length* 5*, we have the following* 6 *possibilities:*
  - $P_{o_2} = \{p_1, p_2, p_3, p_4, p_5\}$, $w(P_{o_2}) = 12$
  - $P_{o_3} = \{p_1, p_2, p_3, p_4, p_6\}$, $w(P_{o_3}) = 13$
  - $P_{o_4} = \{p_1, p_2, p_3, p_5, p_6\}$, $w(P_{o_4}) = 11$
  - $P_{o_5} = \{p_1, p_2, p_4, p_5, p_6\}$, $w(P_{o_5}) = 14$
  - $P_{o_6} = \{p_1, p_3, p_4, p_5, p_6\}$, $w(P_{o_6}) = 15$
  - $P_{o_7} = \{p_2, p_3, p_4, p_5, p_6\}$, $w(P_{o_2}) = 15$

  All of them ensure the observability and P_{o_4} *is kept because has a lower cost.*
- *There are 15 subsets of length* 4*:*
  - $P_{o_8} = \{p_1, p_2, p_3, p_4\}$, $w(P_{o_8}) = 9$

- $P_{o_9} = \{p_1, p_2, p_3, p_5\}$, $w(P_{o_9}) = 7$
- $P_{o_{10}} = \{p_1, p_2, p_3, p_6\}$, $w(P_{o_{10}}) = 8$
- $P_{o_{11}} = \{p_1, p_2, p_4, p_5\}$, $w(P_{o_{11}}) = 10$
- $P_{o_{12}} = \{p_1, p_2, p_4, p_6\}$, $w(P_{o_{12}}) = 11$
- $P_{o_{13}} = \{p_1, p_2, p_5, p_6\}$, $w(P_{o_{13}}) = 9$
- $P_{o_{14}} = \{p_1, p_3, p_4, p_5\}$, $w(P_{o_{14}}) = 11$
- $P_{o_{15}} = \{p_1, p_3, p_4, p_6\}$, $w(P_{o_{15}}) = 12$
- $P_{o_{16}} = \{p_1, p_3, p_5, p_6\}$, $w(P_{o_{16}}) = 10$
- $P_{o_{17}} = \{p_1, p_4, p_5, p_6\}$, $w(P_{o_{17}}) = 13$
- $P_{o_{18}} = \{p_2, p_3, p_4, p_5\}$, $w(P_{o_{18}}) = 11$
- $P_{o_{19}} = \{p_2, p_3, p_4, p_6\}$, $w(P_{o_{19}}) = 12$
- $P_{o_{20}} = \{p_2, p_3, p_5, p_6\}$, $w(P_{o_{20}}) = 10$
- $P_{o_{21}} = \{p_2, p_4, p_5, p_6\}$, $w(P_{o_{21}}) = 13$
- $P_{o_{22}} = \{p_3, p_4, p_5, p_6\}$, $w(P_{o_{22}}) = 14$

Only $P_{o_{10}}$ and $P_{o_{17}}$ do not ensure the observability. According to the first monotonicity property, any subset of places included in these two cannot ensure the observability and so, when they are generated their observability will not be checked. The other ones are possible solutions but the one with lower cost is P_{o_9} and only this one is kept.

- *With 3 places, we have 20 possibilities but only for the following 12 the observability has to be checked:*

  - $P_{o_{23}} = \{p_1, p_2, p_4\}$, $w(P_{o_{23}}) = 7$
  - $P_{o_{24}} = \{p_1, p_2, p_5\}$, $w(P_{o_{24}}) = 5$
  - $P_{o_{25}} = \{p_1, p_3, p_4\}$, $w(P_{o_{25}}) = 8$
  - $P_{o_{26}} = \{p_1, p_3, p_5\}$, $w(P_{o_{26}}) = 6$
  - $P_{o_{27}} = \{p_2, p_3, p_4\}$, $w(P_{o_{27}}) = 8$
  - $P_{o_{28}} = \{p_2, p_3, p_5\}$, $w(P_{o_{28}}) = 6$
  - $P_{o_{29}} = \{p_2, p_4, p_5\}$, $w(P_{o_{29}}) = 9$
  - $P_{o_{30}} = \{p_2, p_4, p_6\}$, $w(P_{o_{30}}) = 10$
  - $P_{o_{31}} = \{p_2, p_5, p_6\}$, $w(P_{o_{31}}) = 8$
  - $P_{o_{32}} = \{p_3, p_4, p_5\}$, $w(P_{o_{32}}) = 10$
  - $P_{o_{33}} = \{p_3, p_4, p_6\}$, $w(P_{o_{33}}) = 11$
  - $P_{o_{34}} = \{p_3, p_5, p_6\}$, $w(P_{o_{34}}) = 9$

  All of them ensure the observability but only the one with minimum cost, i.e., $P_{o_{24}}$, is kept. Observe that for the following subsets the observability will not be checked being subsets of $P_{o_{10}}$ and $P_{o_{17}}$: $\{p_1, p_2, p_3\}$, $\{p_1, p_2, p_6\}$, $\{p_1, p_3, p_6\}$, $\{p_1, p_4, p_5\}$, $\{p_1, p_4, p_6\}$, $\{p_1, p_5, p_6\}$, $\{p_2, p_3, p_6\}$, $\{p_4, p_5, p_6\}$.

- *With 2 places we have to check the observability for the following sets:*
  - $P_{o_{35}} = \{p_2, p_4\}$, $w(P_{o_{35}}) = 6$
  - $P_{o_{36}} = \{p_2, p_5\}$, $w(P_{o_{36}}) = 4$
  - $P_{o_{37}} = \{p_3, p_4\}$, $w(P_{o_{37}}) = 7$
  - $P_{o_{38}} = \{p_3, p_5\}$, $w(P_{o_{38}}) = 5$

  And the following are not checked: $\{p_1, p_2\}$, $\{p_1, p_3\}$, $\{p_1, p_4\}$, $\{p_1, p_5\}$, $\{p_1, p_6\}$, $\{p_2, p_3\}$, $\{p_2, p_6\}$, $\{p_3, p_6\}$, $\{p_4, p_5\}$, $\{p_4, p_6\}$, $\{p_5, p_6\}$. *All sets for which the observability was checked ensure the system observability and we keep only the one with minimum cost,* $P_{o_{36}}$.
- *Taking only one place is not a good choice since all are subsets of* $P_{o_{10}}$ *and* $P_{o_{17}}$.

Hence the optimal solution is $P_{o_{36}} = \{p_2, p_5\}$, *that ensures the system observability at a cost* $w(P_{o_{36}}) = 4$. *Observe that with this brute force algorithm, observability have been checked 38 times.*

4.5.3 Splitting the net in threads

In this section we try to use some properties of contPN systems in order to reduce the complexity of the brute force algorithm presented. We try to reduce the number of observability checks. The idea is to try to group the set of places into subsets such that only one place per subset can belong to an optimal solution. First, since attributions can yield a loss of observability, the net is split in AF parts and the observability is checked in the subnets. Moreover, the number of places from each subnet will be reduced using some properties.

Using Definition 4.20, for an attribution we can define:

Definition 4.43. *Let* $\mathcal{N}$ *be a JC-F contPN and* $p \in P$ *an attribution, i.e.,* $|{}^\bullet p| > 1$. *The AF path from a place* $p'' \in P$ *to* p' *is called a* thread *of* p *if* $p' \in {}^\bullet\left({}^\bullet p\right)$ *and the path is maximal, i.e., there no exists other AF path from a place* p_x *to* p' *that contains the path from* p'' *to* p'.

Therefore, a thread is a maximal connected subnet without attributions (and synchronizations) that "ends" in a place in ${}^\bullet\left({}^\bullet p\right)$ with p an attribution. Since for the observability we are interested in state and output variables (places in our case), an ordered set of places (a list) of the form: $\boldsymbol{g}^l = \left(p_1, \ldots, p_{k+1}\right)$ can be associated to each thread. Abusing notation we will say that $\boldsymbol{g}^l$ is a thread. Moreover, we will try to remove elements from this ordered sets to obtain a lower complexity for the covering problem and we will still call them threads.

Example 4.44. *Let us consider the JF contPN system in Fig. 4.10. Let us compute the threads associated to the attributions. First, there are two attributions, in* p_9 *and* p_{10} *and we obtain:* $\boldsymbol{g}^1 = (p_3, p_1, p_2)$ *and* $\boldsymbol{g}^2 = (p_8, p_6, p_7)$ *for* p_9, *respectively* $\boldsymbol{g}^3 = (p_4, p_3, p_1, p_2)$ *and* $\boldsymbol{g}^4 = (p_9)$ *for* p_{10}.

For $\boldsymbol{g}^l$, we will denote by $\boldsymbol{g}^l(i)$ the i^{th} element of the list. The number h of threads of a JC-F net $\mathcal{N}$ is given by:

$$h = \sum_{|{}^\bullet p|>1} |{}^\bullet p| \tag{4.18}$$

It is obvious that if the first place of each thread is measured, the other ones in the same thread are observed from it. Hence, measuring the first places of all threads and one from

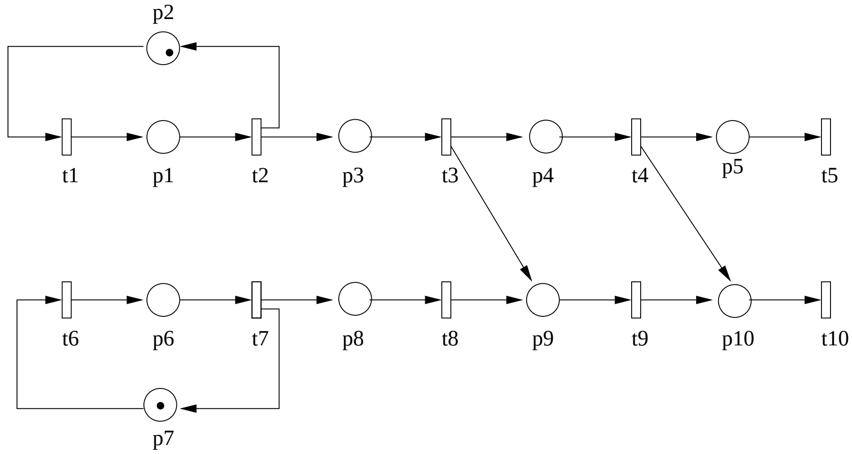


Figure 4.10: A JF conPN system used in Ex. 4.44.

each terminal strongly-connected components, the contPN system will be observable, even structurally observable. But we are interested in minimum cost observability and in general this is not an optimal solution.

Given an attribution place p, algorithm 4.45 computes one of its threads. The input is a place $p_i \in {}^\bullet({}^\bullet p)$ and the returned value is a list $\boldsymbol{g}$ containing all the places that belong to the path in *reversed order*, such that $\boldsymbol{g}(1) = p_i$. To compute all the threads of p, the algorithm should be called for all the places in ${}^\bullet({}^\bullet p)$.

Algorithm 4.45.

procedure THREAD*(p)*
  $\boldsymbol{g} := append(\emptyset, p)$; ▷ *initialize* $\boldsymbol{g}$ *with p*
  while $(|{}^\bullet p| \leq 1) \& ({}^\bullet({}^\bullet p) \notin \boldsymbol{g})$ ***do*** ▷ *until p is not an attrib. and is not already incl. in* $\boldsymbol{g}$
    $p := {}^\bullet({}^\bullet p)$; ▷ *get the backward place*
    $\boldsymbol{g} := append(\boldsymbol{g}, p)$; ▷ *add it to the list*
  end while
  return g
end procedure

Applying Prop. 4.22 it can be seen that at most one place from a thread should be measured in the optimal solution:

Proposition 4.46. *Let* $\boldsymbol{g}$ *be a thread. If* $\boldsymbol{g}(j)$ *is measured then* $\forall k \geq j$, $\boldsymbol{g}(k)$ *is observable.*

Hence, at most one place of the thread belongs to the optimal solution. At this point, the covering problem reduces to trying all combinations of places, taking at most one from each thread. We try now to reduce the number of places in the threads that need to be analyzed while preserving the optimality of the solution.

(Step 1) For every JA-F cycle, choose the place with minimum cost and remove the other places from the threads and the optimal observation is preserved.

If a thread contains a JA-F cycle, in the optimal solution at most one place with minimum cost can appear. A thread contains a cycle iff this cycle is a root strongly-connected component (strongly-connected component that has no input) of the original net.

Proposition 4.47. *Let $\mathcal{N}$ be a JF contPN. If a thread $\boldsymbol{g}$ into an attribution contains a cycle then the cycle is a strongly-connected component of $\mathcal{N}$ without inputs.*

Proof. If it is not a root strongly-connected component then a place or a transition should have two or more inputs. Obviously, none of this is possible because $\boldsymbol{g}$ is JCA-F. □

Example 4.48. *Let us go back to the Example 4.44 where the threads of the net in Fig. 4.10 are given. We will apply the first reduction step. Assume the measurement cost $w = (2,1,1,1,1,1,2,0.5,1,1)$. This net has two root strongly connected components: $\mathcal{S}_1 = \{p_1, p_2\}$ and $\mathcal{S}_2 = \{p_6, p_7\}$. It is clear that knowing p_1, p_2 can be estimated or viceversa. But $w(p_1) = 2 > w(p_2) = 1$, therefore p_1 cannot be in the optimal solution. For $\mathcal{S}_2$, $w(p_6) = 1 < w(p_7) = 2$ hence p_7 cannot be in the optimal solution. Updating $\boldsymbol{g}^1$, $\boldsymbol{g}^2$, $\boldsymbol{g}^3$ and $\boldsymbol{g}^4$, the threads are: $\boldsymbol{g}^1 = (p_3, p_2)$, $\boldsymbol{g}^2 = (p_8, p_6)$, $\boldsymbol{g}^3 = (p_4, p_3, p_2)$ and $\boldsymbol{g}^4 = (p_9)$, respectively. Therefore, the number of places of threads $\boldsymbol{g}^1$, $\boldsymbol{g}^2$ and $\boldsymbol{g}^3$ is reduced.*

(Step 2) Eliminating the places estimated from all terminal strongly connected components.

To further reduce the number of places let us consider the terminal strongly connected components. If at the previous step we remove all the cycles, then at this point, for any terminal strongly connected component (defined in 4.24) one place F_i has been chosen and all of them are essential covers for observability. Therefore, these places and those observable from them can be ignored when the optimal solution is searched. The set of places that are structurally observable from F_i can be computed using the algorithm 4.45 (calling it for the unique place in F_i). The union of all threads, denoted by $\boldsymbol{g}^F$, are the places structurally observable from terminal strongly-connected components.

Example 4.49. *Let us consider the contPN in Fig. 4.10. Its threads have been computed in Ex. 4.44 and have been updated in Step 1, removing some places based on the root strongly connected components in Ex. 4.48. Let us now apply Step 2.*

Let us consider here the estimations due to the terminal strongly connected components. There are two terminal strongly connected components: $F_1 = \{p_5\}$ and $F_2 = \{p_{10}\}$. Hence p_5 and p_{10} will be measured. Using Alg. 4.45, the set of places structurally observable from the terminal strongly connected components is: $\boldsymbol{g}^F = (p_5, p_4, p_3, p_1, p_2, p_{10})$. All of these can be removed from the threads: $\boldsymbol{g}^1 \setminus \boldsymbol{g}^F = \emptyset$, $\boldsymbol{g}^2 \setminus \boldsymbol{g}^F = (p_8, p_6)$, $\boldsymbol{g}^3 \setminus \boldsymbol{g}^F = \emptyset$ and $\boldsymbol{g}^4 \setminus \boldsymbol{g}^F = (p_9)$.

4.5.4 Dominance and an improved algorithm

(Step 3) Eliminating some places in each thread looking at their cost.

Other reduction of the threads can be done considering the measuring cost w. From Prop. 4.46, if $\boldsymbol{g}(j)$ is measured then $\forall k > j$, $\boldsymbol{g}(k)$ is observable. Looking at the cost, if $w(\boldsymbol{g}(k)) \geq w(\boldsymbol{g}(j))$, the place $\boldsymbol{g}(k)$ can be ignored when the solution is searched because for sure will not be in the optimal solution. This represents the *dominance* idea: the number of observable places measuring $\boldsymbol{g}(k)$ is less than or equal to the number of observable places measuring $\boldsymbol{g}(j)$, the cost of $\boldsymbol{g}(k)$ is higher, therefore cannot be in the optimal solution.

After the elimination of the places looking at the cost, the cost observing the places in the reduced thread will be in descending order, i.e., $\forall \mathbf{g}^i$, if $j \leq k$ then $w(\mathbf{g}^i(j)) > w(\mathbf{g}^i(k))$. Thus, in the particular case in which all places have the same measuring cost, each thread will contain only one element and the combinatorial algorithm will have complexity 2^h, where h is the number of threads.

Example 4.50. *Consider the contPN in Fig. 4.10 with the threads given in Ex. 4.44 and updated in Ex. 4.48 and Ex. 4.49 using Step 1 and 2. The obtained threads are:* $\mathbf{g}^1 = \emptyset$, $\mathbf{g}^2 = (p_8, p_6)$, $\mathbf{g}^3 = \emptyset$ *and* $\mathbf{g}^4 = (p_9)$. *Remember that the measurement costs that have been considered are:* $w = (2,1,1,1,1,1,2,0.5,1,1)$.

In the case of $\mathbf{g}^2 = (p_8, p_6)$, $w(p_8) = 0.5 < w(p_6) = 1$. *Since measuring* p_8, p_6 *is observable, for sure* p_6 *cannot be in the optimal solution. Therefore,* $\mathbf{g}^2 = (p_8)$ *and both non-empty threads have only one element. Now, if a covering algorithm is applied, we have to make only* $2^2 = 4$ *combinations, a number that is for sure smaller than the number of combinations of the brute force method. Observe that here we have* 2^h *combinations, not* $2^h - 1$ *as in Ex. 4.42 since here the system can be observable measuring only the terminal strongly connected components that is not the case in Ex. 4.42.*

(Step 4) Making the threads disjoint.

After all these three steps, the number of places that belong to the threads is reduced but may happen that a place p of the net belongs to more that one thread, a fact that will lead to generating more than once some combinations.

Example 4.51. *Let us consider the net in Fig. 4.4(a) that has the following threads:* $\mathbf{g}^1 = (p_2, p_4, p_5)$ *and* $\mathbf{g}^2 = (p_3, p_4, p_5)$. *Making combinations, taking at most one place from each thread,* $\{p_4, p_5\}$ *and* $\{p_5, p_4\}$ *are obtained. Obviously, the observability will be checked only once but the combinations are generated anyhow.*

Therefore, these threads can be made disjoint. This implies that the number of places that belong to at least one of those disjoint threads is less than or at most equal to the number of places in the contPN. In general, to make these threads disjoint many solutions can exist and any of them can be used.

Observe that all the reductions preserve the optimality of the solution. Now, the covering problem can be stated. We have to generate all combinations taking at most one place from each thread and then check the observability of the system. To facilitate the procedure of making of the combinations, a null element will be inserted at the end of each thread. If the system is observable, the solution is kept if has a cost lower than the previous one. A good choice is starting with the first places of each thread and going backward since Prop. 4.46 can be used in the following way: if the system is not observable for the current observations we don't have to advance in the threads because the system will not be observable.

Example 4.52. *Let us go back to the net in Fig. 4.9 with* $\boldsymbol{\lambda} = \mathbf{1}$ *and* $w = (1,1,2,5,3,4)$. *The brute force algorithm was applied in Ex. 4.42 checking for the observability 38 times. Let us apply the technique of threads.*

This net has the following threads: $\mathbf{g}^1 = (p_3, p_1)$, $\mathbf{g}^2 = (p_2)$, $\mathbf{g}^3 = (p_5, p_1)$ *and* $\mathbf{g}^4 = (p_4, p_6, p_1)$. *Applying the reductions of the threads, Step 1,2 and 3 do not modify them since after Step 4, a solution can be:* $\mathbf{g}^1 = (p_3, p_1, \emptyset)$, $\mathbf{g}^2 = (p_2, \emptyset)$, $\mathbf{g}^3 = (p_5, \emptyset)$ *and* $\mathbf{g}^4 = (p_4, p_6, \emptyset)$.

The combinations obtained are:

- $P_{o_1} = \{p_3, p_2, p_5, p_4\}$ *(first place of each thread) that ensures the system observability at a cost* $w(P_{o_1}) = w_3 + w_2 + w_5 + w_4 = 11$

- $P_{o_2} = \{p_3, p_2, p_5, p_6\}$ *(advancing in* $\mathbf{g}^4$*) ensures the observability at a cost* $w(P_{o_2}) = 10$ *and becomes a possible solution replacing* P_{o_1}

- $P_{o_3} = \{p_3, p_2, p_5\}$ *(advancing in* $\mathbf{g}^4$*) that is kept because ensure the observability at a cost* $w(P_{o_3}) = 6$

- $P_{o_4} = \{p_3, p_2, p_4\}$ *we obtain an observable system at a cost* $w(P_{o_4}) = 8$ *but it is higher than* $w(P_{o_3})$

- *with* $P_{o_5} = \{p_3, p_2, p_6\}$*, the system is not observable and according to Prop. 4.46 it is not necessary to keep advancing in the threads from this node since the system will not be observable. Therefore, for the following combinations the observability will not be generated:* $\{p_3, p_2\}$, $\{p_3, p_6\}$, $\{p_3\}$, $\{p_1, p_2, p_6\}$, $\{p_1, p_2\}$, $\{p_1\}$, $\{p_2, p_6\}$, $\{p_2\}$ *and* $\{p_6\}$

- $P_{o_6} = \{p_3, p_5, p_4\}$ *with* $w(P_{o_6}) = 10$ *ensures the system observability but* $w(P_{o_6}) > w(P_{o_3})$

- $P_{o_7} = \{p_3, p_5, p_6\}$ *with* $w(P_{o_7}) = 9$ *ensures the system observability but* $w(P_{o_7}) > w(P_{o_3})$

- $P_{o_8} = \{p_3, p_5\}$ *with* $w(P_{o_8}) = 5$ *ensures the system and will become the candidate to the optimal solution since* $w(P_{o_8}) < w(P_{o_3})$

- *next node is* $P_{o_9} = \{p_3, p_4\}$ *with* $w(P_{o_9}) = 7$

- $P_{o_{10}} = \{p_1, p_2, p_5, p_4\}$ *with* $w(P_{o_{10}}) = 10$ *ensures the system observability but* $w(P_{o_{10}}) > w(P_{o_8})$

- $P_{o_{11}} = \{p_1, p_2, p_5, p_6\}$ *with* $w(P_{o_{11}}) = 9$ *ensures the system observability but* $w(P_{o_{11}}) > w(P_{o_8})$

- $P_{o_{12}} = \{p_1, p_2, p_5\}$ *with* $w(P_{o_{12}}) = 5$ *ensures the system observability and* $w(P_{o_{11}}) = w(P_{o_8})$*, hence it is also a possible solution*

- $P_{o_{13}} = \{p_1, p_2, p_4\}$ *with* $w(P_{o_{13}}) = 7$ *ensures the system observability but* $w(P_{o_{13}}) > w(P_{o_8})$

- *measuring* $P_{o_{14}} = \{p_1, p_5, p_4\}$ *the system is not observable, hence is not necessary to advance in threads from this node. Consequently, for the following threads, the observability will not be checked:* $\{p_1, p_5, p_6\}$, $\{p_1, p_5\}$, $\{p_5, p_4\}$, $\{p_5, p_6\}$, $\{p_5\}$ *and* $\{p_6\}$

- $P_{o_{15}} = \{p_2, p_5, p_4\}$ *with* $w(P_{o_{15}}) = 9$ *ensures the system observability but* $w(P_{o_{14}}) > w(P_{o_8})$

- $P_{o_{16}} = \{p_2, p_5, p_6\}$ *with* $w(P_{o_{16}}) = 8$ *ensures the system observability but* $w(P_{o_{16}}) > w(P_{o_8})$

- $P_{o_{17}} = \{p_2, p_5\}$ *with* $w(P_{o_{17}}) = 4$ *ensures the system and may be the optimal solution since* $w(P_{o_{17}}) < w(P_{o_8})$

- *finally,* $P_{o_{18}} = \{p_2, p_4\}$ *with* $w(P_{o_{18}}) = 6$ *but* $w(P_{o_{18}}) > w(P_{o_{17}})$

Then, the optimal solution is $P_{o_{17}}$ *and its cost is* 4*. Observe that we have obtained the same solution as in brute force algorithm but now, the number of observability checks is 18. This represent a drastically reduction in comparison with brute force algorithm where 38 checks of the observability are needed.*

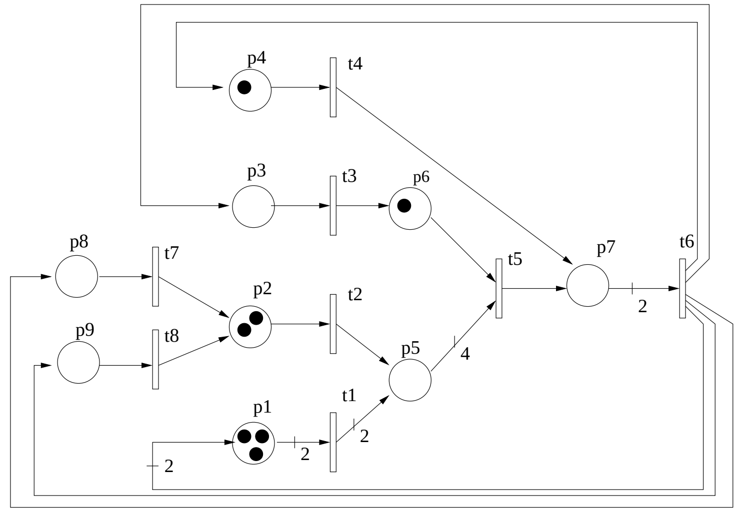


Figure 4.11: Petri net modeling a table factory.

The results presented are combined in Algorithm 4.53 to obtain the optimal solutions for observability in the case of a given $\boldsymbol{\lambda}$. The complexity of the algorithm remains exponential, but in practice has been considerably reduced.

Algorithm 4.53.

procedure OPT-OBSERV*($\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$)*
    if *$\mathcal{N}$ is AF* ***then***
        remove all joins (Prop. 4.27).
    end if
    remove all the synchronizations/joins that are in CEQ (Prop. 4.30).
    if *$\mathcal{N}$ has more joins* ***then***
        return *error: the algorithm cannot be applied*
    end if
    Compute the terminal str.connected components F and $\mathbf{g}^F$ *executing Alg. 4.45.*
    if *∄ attributions (Prop. 4.25 applies)* ***then***
        return *one place from each term. str. conn. comp. with minimum cost and the cost.*
    else
        Compute the threads $\mathbf{g}^j$
        Step 1*: remove the cycles from threads*
        Step 2*: remove the str. obsv. places from terminal comp.*
        Step 3*: update the threads looking at their costs.*
        Step 4*: make the threads disjoint.*
        solve the covering problem
    end if
end procedure

Example 4.54. *The small net system presented in Fig. 4.11 models a FMS that consists of three different machines to make table-legs, one (t_1) which produces two legs at a time, and two (t_7 and t_8) which make legs one by one; one machine (t_3) to produce the table-boards; one* *machine (t_5) to assemble four legs and a board; And a big painting line (t_6) which paints two tables at once. The painting line has more capacity than the other machines, so more unpainted tables are brought (t_4) from a different factory. The different products are stored in buffers: Table-legs are stored in p_5, the ones produced by the slow machines are first stored in p_2 and have to be taken to p_5 (operation t_2), boards are stored in p_6, and p_7 is devoted to the storage of unpainted tables. The rest of places contains work orders: Whenever the painting line finishes a couple of tables, it delivers work orders to the leg-makers, the board-maker, and the other factory. Moreover, 50% of the tables are assembled, and 50% are brought from the other factory, while 50% of the legs are produced by the fast leg-maker, and 50% by the slow one, half and half.*

The number of clients in each buffer can be measured with a sensor, the cost of the sensor depends on the buffer. We want to know the minimum amount of money that we should invest to make the system observable.

Let us apply the Alg. 4.53 assuming $\boldsymbol{\lambda} = \mathbf{1}$ and $w = (1,1,1,1,1,1,1,1,1)$. After removing the synchronization, $F_1 = \{p_5\}$, $F_2 = \{p_6\}$ and $\mathbf{g}^F = (p_5, p_6, p_3, p_7, p_4)$. Measuring p_5 and p_6 the system is not observable. Then, we need to compute the threads *that are: $\mathbf{g}^1 = (p_1, p_7)$, $\mathbf{g}^2 = (p_2)$, $\mathbf{g}^3 = (p_8, p_7)$ and $\mathbf{g}^4 = (p_9, p_7)$. Applying first step of the reductions, these threads are not changed since the net has no root strongly connected component. In Step 2, we obtain: $\mathbf{g}^1 \setminus \mathbf{g}^F = (p_1)$, $\mathbf{g}^2 \setminus \mathbf{g}^F = (p_2)$, $\mathbf{g}^3 \setminus \mathbf{g}^F = (p_8)$ and $\mathbf{g}^4 \setminus \mathbf{g}^F = (p_9)$ that remain unchanged after Step 3 and 4. Applying the covering problem, the following sets of places are considered and the observability will be checked for them: $P_{o_1} = \{p_1, p_2, p_8, p_9\}$, $P_{o_2} = \{p_2, p_8, p_9\}$, $P_{o_3} = \{p_1, p_8, p_9\}$, $P_{o_4} = \{p_1, p_2, p_9\}$, $P_{o_5} = \{p_1, p_2, p_8\}$, $P_{o_6} = \{p_8, p_9\}$, $P_{o_7} = \{p_2, p_9\}$, $P_{o_8} = \{p_2, p_8\}$, $P_{o_9} = \{p_1, p_9\}$, $P_{o_{10}} = \{p_1, p_8\}$ and $P_{o_{11}} = \{p_1, p_2\}$. The optimal solution is one of: P_{o_7}, P_{o_8}, P_{o_9} or $P_{o_{10}}$.*

Observability was checked only 12 times applying the algorithm that it is much less than solving with brute force method.

We have considered that only the marking of the places can be measured, but measuring the flow through transitions might be possible. When a non join transition is measured, based on the infinite server semantics that it is considered there, the marking of the input place is computed immediately. Hence, measuring the flow through the transition or the marking of the input place is equivalent. If a transition is a join, measuring the flow through it, the enabling degree of the transition is estimated. In general, the marking of its input places cannot be computed from the minimum. Anyhow, part of the previous results obtained measuring the instantaneous marking of some places can be easily extended when the flow of the transitions is measured or in a mixed case (measuring the markings of some places and the flow through some transitions).

Chapter 5

State estimation of continuous Petri nets with finite server semantics

Summary

In this chapter we consider the state estimation problem for untimed nets and timed nets with finite server semantics, trying to determine all states that are consistent with an observed sequence of transition firings. Firstly, we show how the results previously obtained for discrete nets can be applied, with minor modifications, to untimed continuous net systems. Secondly, we consider timed continuous Petri nets with a particular relaxation of finite server semantics. Under the assumption that no measuring is available (thus the set of consistent markings only depends on the time elapsed), we study the observation based on the time-reachability analysis.

5.1 Motivation

In this chapter we will study the state estimation problem of continuous system under finite server semantics. We assume that the initial marking is known and based on some observations of the flows we want to determine the set of markings that are consistent with the observation.

Finite server semantics is taken with a particular relaxation: transitions are not forced to fire at their maximum firing speeds but can fire with a speed smaller than its maximum. This semantics it is used for the continuous part of so called First Order Hybrid Petri Nets (FOHPN) [7] and we will show in Section 5.5 that can be seen as finite server semantics defined in [2] with control.

We consider the observation problem for both untimed continuous nets and timed contPN with finite server semantics under this relaxation. Following [25], we make the following assumptions:

(A1) the initial marking $\boldsymbol{m}_0$ is known;

(A2) the net structure is known.

(A3) the set of transitions is partitioned into $T = T_o \cup T_u$, where T_o is the set of *observable transitions*, whose firing is known during all time trajectory and T_u is the set of *unobservable* or *silent transitions*, whose firing cannot be measured directly.

First, let us see that the results in chapter 4 cannot be easily extended to finite server semantics. Under this semantics, the flow of a transition t_i is defined by equations (2.7). So, it is constant and equal with its *maximum firing speed* when all input places *contain tokens/fluid* (their markings are positive) and it is the minimum input flow if there are input places without tokens/fluid.

The algorithms in the previous chapter are based on the going backward procedure: using the flow of a transition, the marking of its input places is estimated. In this case, it is easy to see that the flow of one transition cannot provide, in general, enough information to estimate the marking of the input places. Therefore, it is not possible to extend the algorithm presented in section 4.5 for infinite server semantics to finite server semantics.

Let us consider the contPN system in Figure 5.1 and assume finite server semantics with $\boldsymbol{\lambda} = [0.5, 0.5]^T$. For the initial marking in the figure, the flow of the transitions will be $\boldsymbol{f} = [0.5, 0.5]^T$ (t_2 will fire at the maximum speed because $m_1 > 0$ and $f_2 = \min\{\lambda_2, 0.5\} = 0.5$). For the contPN system with $\boldsymbol{m}_0 = [3, 0]^T$, the same flow will be obtained.

In the case of untimed contPN the nets behavior is asynchronous and sequential, as in discrete nets with interleaving semantics; the only difference between the first and the latter model is the relaxation of the integer constraint (see section 2.2). In this case after each observable transition fires we observe its firing quantity which is the continuous counterpart of the number of firings. The set of markings consistent with an observation σ_o, i.e., the set of markings in which the net may be after a firing sequence σ_o is observed, will be denoted $\mathcal{T}(\sigma_o)$. It is not surprising that most of the results derived for discrete nets also apply in this case. The state estimation problem for this class of nets is briefly discussed in Section 5.2.

In the case of timed continuous nets the situation is significantly different for two main reasons. Firstly, transitions may fire in parallel and what we observe is the instantaneous firing speed of observable transitions. Secondly, timing constraints must be taken into account and embedded into the state estimation procedure.

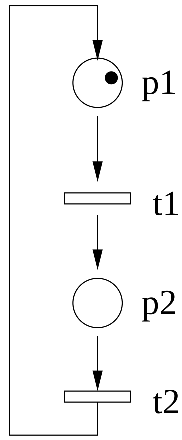


Figure 5.1: A simple contPN.

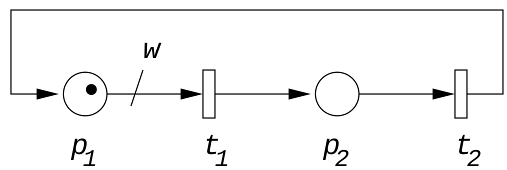


Figure 5.2: ContPN system for which the marking $[0,0]^T$ is lim-reachable in the untimed system but reachable in the timed one with finite server semantics if $w = 2$.

For example, let us consider the net in Fig. 5.2 with arc weight $w = 1$, where the instantaneous firing speed of each transition must belong to the interval $[0, 1]$. Assume that the observed flow of transition t_2 is $v_2(\tau) = 0.5$ during a time interval $[0, 0.5]$, while the flow v_1 of transition t_1 cannot be observed. We want to determine the marking consistent with this observation, given that it holds that $m_1(\tau) = 1 - (v_1 - v_2) \cdot \tau$ and $m_2(\tau) = (v_1 - v_2) \cdot \tau$. Obviously, we impose the non-negativity conditions on m_1 and m_2, and also observe that m_1 cannot empty until 0.5 t.u. Since t_2 is firing with firing speed 0.5, to keep the marking of p_2 non negative, transition t_1 must have been firing in parallel during this time interval, with an average speed of at least 0.5. However, t_1 may be firing with an even greater speed, up to $v_1 = 1$; thus the set of consistent markings in the considered observation interval is:

$$\mathcal{T}(v_2(\cdot), \tau) = \{[1 - m,\ m]^T \mid 0 \le m \le 0.5\tau\}.$$

This shows that the set of consistent markings explicitly depends not only on the observed firing speeds but also on the elapse of time.

A first approach to state estimation of timed continuous nets with particular relaxation of finite server semantics is presented in the second part of the chapter. We assume that no observation is available, thus the observation problem reduces to determining the set of markings $\mathcal{T}(\tau)$, in which the net may be at time τ. This problem is similar to that of time-reachability for continuous models: this is why in Section 5.4 we also study the equivalence of reachability of the continuous untimed model and reachability of the timed one showing under which conditions it holds. For some classes, a procedure to compute the minimum time such that the set of the consistent markings is the same as the reachability space is given.

5.2 State estimation of untimed contPN

First, we concentrate on the state estimation of untimed contPN following [25]. As mentioned in Assumption (A3) we consider that the set of transitions is divided into $T = T_o \bigcup T_u$, where T_o is the set of observable transitions and T_u is the set of unobservable transitions. We assume that the firing count vector $\boldsymbol{\sigma} \in \mathbb{R}^n_{\geq 0}$ has two components: $\boldsymbol{\sigma}_o \in \mathbb{R}^{n_o}_{\geq 0}$ associated to the observable transitions and $\boldsymbol{\sigma}_u \in \mathbb{R}^{n_u}_{\geq 0}$ associated to the unobservable transitions, where n, n_o and n_u are respectively the corresponding sets.

Let us introduce the following definitions:

Definition 5.1. *Given a vector* $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_o^T\ \boldsymbol{\sigma}_{uo}^T]^T \in \mathbb{R}^n_{\geq 0}$*, we define its observable projection as:*

$$\Gamma(\boldsymbol{\sigma}) = \boldsymbol{\sigma}_o$$

■

This definition can also be applied to a firing sequence σ.

Definition 5.2. *Given an untimed contPN* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *we define* $\mathcal{N}_u$ *the net obtained from* $\mathcal{N}$ *removing all observable transitions.* ■

As shown in the introduction, observing the firing of an observable transition may allow to reconstruct the firing of a transition that is not observable. To characterize this fact we introduce the following definitions.

Definition 5.3. *Given a marking* $\boldsymbol{m}$*, an observable transition* $t \in T_o$*, and* $\alpha > 0$*, we define the set of* explanations *of* $t(\alpha)$ *at* $\boldsymbol{m}$ *as:*

$$\Sigma(\boldsymbol{m}, t(\alpha)) = \{\sigma_u \in \mathcal{L}(\mathcal{N}_u, \boldsymbol{m}_0)\ \mid \boldsymbol{m}[\sigma_u\rangle \boldsymbol{m}',\ \boldsymbol{m}' \geq \alpha \boldsymbol{Pre}(\cdot, t)\}$$

where $\mathcal{L}(\mathcal{N}, \boldsymbol{m}_0)$ *is the set of all fireable sequences in the net. We denote*

$$Y(\boldsymbol{m}, t(\alpha)) = \{\boldsymbol{\sigma}_u \in \mathbb{R}^{n_u}_{\geq 0}\ \mid \sigma_u \in \Sigma(\boldsymbol{m}, t(\alpha))\}$$

the corresponding set of firing vectors. ■

Thus $\Sigma(\boldsymbol{m}, t(\alpha))$ is the set of unobservable sequences whose firing is necessary to enable $t(\alpha)$. Among the above vectors we want to select those whose firing vector is minimal, that we call *minimal e-vectors.*

Definition 5.4. *Given a marking* $\boldsymbol{m}$ *and an observable transition* $t \in T_o$ *with its firing amount* α*, we define the set of* minimal explanations *of* $t(\alpha)$ *at* $\boldsymbol{m}$*:*

$$\Sigma_{min}(\boldsymbol{m}, t(\alpha)) = \{\sigma_u \in \Sigma(\boldsymbol{m}, t(\alpha)) \mid \nexists\ \sigma_u' \in \Sigma(\boldsymbol{m}, t(\alpha)) : \boldsymbol{\sigma}_u' \leq \boldsymbol{\sigma}_u\}$$

and we denote

$$Y_{min}(\boldsymbol{m}, t(\alpha)) = \{\boldsymbol{\sigma}_u \in \mathbb{R}^{n_u}_{\geq 0} \mid\ \sigma_u \in \Sigma_{min}(\boldsymbol{m}, t(\alpha))\}$$

the corresponding set of e-vectors. ■

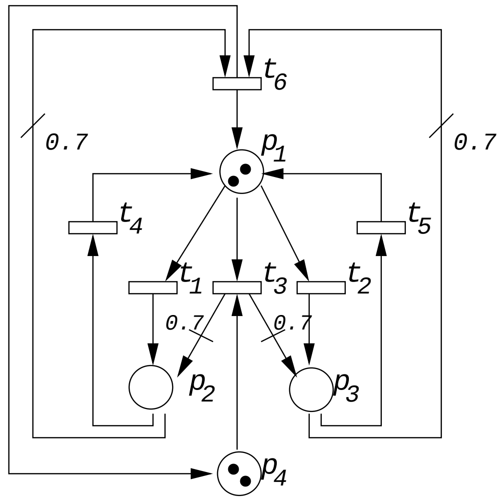


Figure 5.3: Conservative and consistent contPN used in Ex. 5.5 and Ex. 5.24.

Example 5.5. *Let us consider the contPN in Fig. 5.3 with* $T_o = \{t_4, t_5, t_6\}$ *and* $T_u = \{t_1, t_2, t_3\}$. *Assume the initial marking in the figure and an observation of the firing of* t_6 *in an amount* 0.7. *Observe that this firing is not possible at this marking since the input places in* t_6, p_1 *and* p_2, *are empty initially. This means that some unobservable transitions were fired before the firing of* t_6.

To fire $t_6(0.7)$ *places* p_2 *and* p_3 *should be marked at least with* 0.49 *tokens. To put* 0.49 *tokens in* p_2 *and* p_3, t_1 *and* t_2 *should fire in an amount* 0.49 *or* t_3 *should fire in an amount* 0.7. *Obviously, a combination of the firings of* t_1, t_2 *and* t_3 *is also possible. Hence,*

$$\Sigma_{min}(\boldsymbol{m}_0, t_6(0.7)) = \left\{\alpha \cdot 0.49 t_1 t_2 + \beta \cdot 0.7 t_3 | \alpha + \beta = 1, \alpha, \beta \geq 0\right\}$$

Note that if $t(\alpha)$ is enabled at $\boldsymbol{m}$, then $\Sigma_{min}(\boldsymbol{m}, t(\alpha)) = \emptyset$. Now let us introduce an algorithm to compute the firing vectors of a set of minimal explanations. In the discrete case, the algorithm returns all sets of minimal explanations. In the continuous case, if the returned value of the algorithm is not unique then all combinations of the vectors obtained are also minimal explanations (see the previous example), hence an infinite number of solutions are obtained.

Algorithm 5.6. ***procedure*** COMPUTATION OF $Y_{\min}(\boldsymbol{m}, t(\alpha))$

Let $\Gamma := \left|\begin{array}{c|c} C_u^T & I_{n_u \times n_u} \\ \hline A & B \end{array}\right|$, *where* $A := (\boldsymbol{m} - \alpha Pre[\cdot, t])^T$, $B := \vec{0}_{n_u}^T$.

if $A < 0$ ***then***

Choose an element $A[i^*, j^*] < 0$.

Let $\mathscr{I}^+ = \{i \mid C_u^T[i, j^*] > 0\}$.

for *all* $i \in \mathscr{I}^+$ ***do***

add to $[A \mid B]$ *a new row* $[A[i^*, \cdot] + kC_u^T[i, \cdot] \mid B[i^*, \cdot] + k\boldsymbol{e}_i^T]$,

where $\boldsymbol{e}_i$ *is the* i*-th canonical basis vector and* $k = -\frac{A[i^*, j^*]}{C_u^T[i, j^*]}$.

$\triangleright$ *the element* $A[i^*, j^*]$ *is now set to 0*

end for

Remove the row $[A[i^*, \cdot] \mid B[i^*, \cdot]]$

end if
Remove from B any row that covers other rows.
Each row of B is a vector in $Y_{\min}(\boldsymbol{m}, t(\alpha))$.
end procedure

In previous algorithm, C_u is the restriction to the unobservable transitions of the incidence matrix. Vector A contains the marking obtained after firing only the observable transitions, that can have negative components. To avoid this, a set of unobservable transitions should fire, for this reason we consider only C_u. The algorithm stops when $A \geq 0$. It is similar to the one presented in [35] with two main differences. The first difference is that we have to consider not only that a transition has fired, but also we have to take into account its firing amount. The second difference is in the part in which we have to make linear combinations of rows, where we can simply set to zero the element $A[i^*, j^*]$ without taking into account the integer constraint.

As mentioned, if the set of minimal explanations obtained in Alg. 5.6 is not a singleton then we have infinite number of minimal explanations.

Proposition 5.7. *Let* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *be an untimed contPN system. If its* $\mathcal{N}_u$ *is acyclic and AF, then* $|Y_{min}(\boldsymbol{m}, t(\alpha))| = 1$.

Proof. It is proved in [25] that for discrete nets with $\mathcal{N}_u$ acyclic and AF, for all $\boldsymbol{m}$ and for all firing transitions $t(\alpha)$, $|Y_{min}(\boldsymbol{m}, t(\alpha))| = 1$ (Theorem 4). This result holds also in continuous case since the same definition is used for $Y_{min}(\boldsymbol{m}, t(\alpha))$. □

Under the conditions of Prop. 5.7, we can define the *basis marking* $\boldsymbol{m}_b(\sigma_o)$, as the marking reached from $\boldsymbol{m}_0$ by firing the sequence σ_o and its minimal explanation. The basis marking can be computed recursively. Observe that $\boldsymbol{m}_b(\varepsilon) = \{\boldsymbol{m}_0\}$, i.e., when no observation has occurred, (i.e., $\sigma_o = \varepsilon$), the set of basis marking is equal to the initial marking. Moreover, $\boldsymbol{m}_b(\sigma_o t_o(\alpha)) = \boldsymbol{m}_b(\sigma_o) + C_u \bar{\boldsymbol{\sigma}}_u + \alpha C[\cdot, t_o]$, where $\boldsymbol{m}' = \boldsymbol{m}_b(\sigma_o) + \alpha C[\cdot, t_o]$ and $\bar{\boldsymbol{\sigma}}_u = Y_{min}(\boldsymbol{m}', t_o(\alpha))$, i.e., the basis marking in which the system can be after $\sigma_o t_o(\alpha)$ is reached from the previous basis marking $\boldsymbol{m}_b(\sigma_o)$ with the firing of the observed transitions at $\boldsymbol{m}_b(\sigma_o)$ and the e-vector $\bar{\boldsymbol{\sigma}}_u$.

Finally, given an observable firing sequence σ_o, we can define a set of σ_o-consistent markings as the set of all markings in which the system may be after the sequence σ_o.

Definition 5.8. *Given a firing sequence* σ_o *we define*

$$\mathcal{T}(\sigma_o) = \{ \quad \boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|} \mid (\exists \sigma \in \mathcal{L}(\mathcal{N}, \boldsymbol{m_0})) \\ \Gamma(\sigma) = \sigma_o \ \wedge \ \boldsymbol{m}_0[\sigma\rangle \boldsymbol{m}\}$$

the set of σ_o-consistent markings. ■

We can also give an algebraic characterization of the set of the marking consistent with the observation σ_o for AF systems. Since in contPN some markings can be lim-reachable the set of σ_o-consistent markings should consider this case and these markings are lim-reachable.

Proposition 5.9. *Let* $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ *be an contPN system with its* $\mathcal{N}_u$ *AF. Given a firing sequence* σ_o *the set of* σ_o- consistent markings *is*

$$\mathcal{T}(\sigma_o) = \{\boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|} \mid (\exists \sigma_u \in \mathcal{L}(\mathcal{N}_u, \boldsymbol{m}_b)) \quad \boldsymbol{m} = \boldsymbol{m}_b(\sigma_o) + C_u \boldsymbol{\sigma_u}\}$$

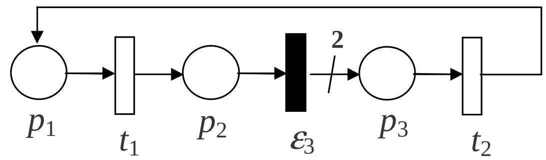


Figure 5.4: Untimed contPN used in Example 5.10

■

Proof. This result follows from the definition of basis marking as in the previous work for discrete nets (Theorem 11 in [25]) and from Theorem 2.10 because $\sigma_u \in \mathcal{L}(\mathcal{N}_u, \boldsymbol{m}_b) \implies \|\boldsymbol{\sigma}_u\| \in FS(\mathcal{N}_u, \boldsymbol{m}_b)$ and implies that $\boldsymbol{m}$ is lim-reachable from $\boldsymbol{m}_b(\sigma_o)$. □

Example 5.10. *Consider the net in Fig. 5.4, where $T_o = \{t_1, t_2\}$ and $T_u = \{\varepsilon_3\}$. Given the marking $\boldsymbol{m} = [0,1,0]^T$ and the transition $t_2(0.5)$, the set of minimal explanations is $\Sigma(\boldsymbol{m}, t_2(0.5)) = \varepsilon_3(0.25)$. Given the marking $\boldsymbol{m} = [1,0,0]^T$ and the observable firing sequence $\sigma_o = t_1(1)$ the basis marking $\boldsymbol{m}_{b,\sigma_o} = \{[0,1,0]^T\}$ and the set of consistent markings according to Prop. 5.9 is $\mathcal{T}(\sigma_o) = \{[0, \beta, 2(1-\beta)]^T\}$, where β can assume any real value between* 0 *and* 1.

5.3 Relaxing finite server semantics

In this chapter we consider the continuous part of the First Order Hybrid Petri Nets [7] that can be seen as a particular relaxation or controlled finite server semantics.

Definition 5.11. *A timed contPN system $\langle \mathcal{N}, \boldsymbol{m}_0, \mathcal{V} \rangle$ with relaxed finite server semantics is a contPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ together with a function $\mathcal{V} : T \to \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}$ that associates to each transition t_j a firing interval $\mathcal{V}(t_j) = [V_m^j, V_M^j]$.* ■

In the untimed case, a contPN evolves sequentially and only one transition is fired at a time. When time is present, more than one transition can be fired at the same time. There are two types of enabling: *strong* and *weak enabling*.

A transition t_j is *strongly enabled* if $\forall p_i \in {}^\bullet t_j$, $m_i > 0$. When $\exists p_i \in {}^\bullet t_j$ such that $m_i = 0$, then t_j is *weakly enabled* iff the empty input places are feed by other transitions. If an empty input place cannot receive input flow then the transition is not enabled.

Observe that we consider the same notion of enabling given in [2], that is different from the one used in [7]. The notion used in [2] prevents the firing of transitions that belong to an empty cycle. See Section 4.3. in [6] for more details. However, considering the same notion of enabling as in [2] does not imply that the flow of the transitions is computed by the same formula. In fact, here the model is more general since the flow can take a smaller value.

The firing interval $[V_m^j, V_M^j]$, associated to the transition $t_j \in T$ through the function $\mathcal{V}$ has the following interpretation: V_m^j represents the *minimum firing speed* at which t_j can fire and V_M^j represents the *maximum firing speed* at which t_j can fire.

At a marking $\boldsymbol{m}$, the *instantaneous firing speed* (IFS) (or the flow) of a transition t_j, denoted v_j is given by:

- if t_j is not enabled then $v_j = 0$;
- if t_j is strongly enabled then it may fire with any firing speed $v_j \in [V_m^j, V_M^j]$;
- if t_j is weakly enabled then it may fire with any firing speed $v_j \in [V_m^j, \bar{V}^j]$, where

$$\bar{V}^j = \min\left\{ \min_{p_i \in {}^\bullet t_j | m_i = 0} \left\{ \sum_{t_k \in {}^\bullet p_i} \frac{v_k \cdot \boldsymbol{Post}[t_k, p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}, V_M^j \right\} \tag{5.1}$$

The value $\bar{V}^j$ in (5.1), corresponding to a weak enabled transition t_j, is computed in such way that the marking of the input places of t_j that are empty will not become negative. Hence, the flow of t_j depends on the input flows in the empty input places. If the input flow is greater than V_M^j then the flow is bounded by this value. We assume that the net is well defined, such that $\bar{V}^j \geq V_m^j$ for all reachable markings. Observe that in the case of $V_m^j = 0$ the net is well defined.

The instantaneous firing speed is piecewise constant. It remains constant until a *macro-event* happens. We have two types of macro-events: (1) *internal macro-events* appearing when a place becomes empty and a new flow-computation is required to ensure the non-negativity of the markings and, (2) *external macro-events* appearing when the external operator change the IFS of some transitions. Therefore, a timed contPN is a piecewise constant system and the period in which the IFS is constant is called *macro-period*.

A procedure to compute the set of admissible IFS vectors at $\boldsymbol{m}$ is given in [7] based on a set of linear equations and inequations. Let $\boldsymbol{v}$ be a feasible solution of the following linear set:

$$\begin{cases} v_j \leq V_M^j & \forall t_j \in T \\ v_j \geq V_m^j & \forall t_j \in T \\ \boldsymbol{C}[p, \cdot] \cdot \boldsymbol{v} \geq 0 & \forall p \in P \text{ with } \boldsymbol{m}[p] = 0 \end{cases} \tag{5.2}$$

The first two equations in (5.2) correspond to the bounds of the firing IFS that should be respected by all transitions (strongly and weakly enabled), while the last equation corresponds to (5.1). Since transitions that belong to an empty cycle cannot be fired, we have to remove from the previous solutions those containing transitions that belong to empty cycles. Let $\mathscr{S}(\mathcal{N}, \boldsymbol{m})$ be the set of all admissible IFS vectors at marking $\boldsymbol{m}$.

Example 5.12. *Let us go back to the Ex. 2.25 (system of Fig. 2.3) but assuming now a relaxed finite server semantics with* $\mathcal{V}(t_1) = [0, 1]$, $\mathcal{V}(t_2) = [0, 2]$, $\mathcal{V}(t_3) = [0, 1]$, $\mathcal{V}(t_4) = [0, 1]$ *and* $\mathcal{V}(t_5) = [0, 0.5]$.

At $\tau = 0$ *since* $\boldsymbol{m}_0 = [1, 1, 0, 0, 0, 1]^T$ *then* t_1 *and* t_4 *are strongly enabled and can fire each one with any firing speed in the interval* $[0, 1]$. *Let us assume that* $v_1 = 0.7$ *and* $v_4 = 0.9$ *are chosen. Transition* t_2 *is weakly enabled hence can be fired with any speed between* 0 *and* 0.7*, the input flow in the input empty place* p_3 *(i.e.* v_1*). Assume it is fired with* $v_2 = 0.5$. *The possible firing speed for* t_3 *is upper bounded by* 0.5*, the input flow in* p_4 *since it is weakly enabled. Assume the maximum firing speed is chosen, i.e,* $v_3 = 0.5$. *The instantaneous firing speed of* t_5 *is at most* 0.5*, the maximum firing speed even if it is weakly enabled because the input flow in* p_5 *is* $v_4 = 0.9$. *Let* $v_5 = 0.5$.

Then, the system will start evolving using: $\boldsymbol{v} = [0.7, 0.5, 0.5, 0.9, 0.5]^T$ *and the marking evolution is given by:*

$$\Sigma_1 = \begin{cases} \dot{m}_1 = -v_1 + v_3 = -0.2 \\ \dot{m}_2 = -v_4 + v_5 = -0.4 \\ \dot{m}_3 = -v_2 + v_1 = 0.2 \\ \dot{m}_4 = -v_3 + v_2 = 0 \\ \dot{m}_5 = -v_5 + v_4 = 0.4 \\ \dot{m}_6 = -v_2 - v_4 + v_3 + v_5 = -0.4 \end{cases} \tag{5.3}$$

This flow can be kept constant if no external event appears until $\tau = 2.5$ *when places* p_2 *and* p_6 *will become empty and an internal event will appear. Let us assume that at* $\tau = 1$*, an external operator changes the firing speed of transition* t_1 *from* $v_1 = 0.7$ *to a new value* $v_1 = 0.1$*. At this time, according to* (5.3) *the current marking is:* $\boldsymbol{m}_1 = [0.8, 0.6, 0.2, 0, 0.4, 0.6]^T$*. From that moment, the system will evolve according to other system, given by:*

$$\Sigma_2 = \begin{cases} \dot{m}_1 = -v_1 + v_3 = 0.4 \\ \dot{m}_2 = -v_4 + v_5 = -0.4 \\ \dot{m}_3 = -v_2 + v_1 = -0.4 \\ \dot{m}_4 = -v_3 + v_2 = 0 \\ \dot{m}_5 = -v_5 + v_4 = 0.4 \\ \dot{m}_6 = -v_2 - v_4 + v_3 + v_5 = -0.4 \end{cases} \tag{5.4}$$

An internal event appears at $\tau = 1+0.5 = 1.5$ *when place* p_3 *empties (assuming no other external event happens before). At this time, the current marking will be:* $\boldsymbol{m}_2 = [1, 0.4, 0, 0, 0.6, 0.4]^T$ *and a new flow computation is required to ensure the positiveness of the markings. Now,* t_1 *and* t_5 *are strongly enabled and can be fired with any speed between* $[0,1]$ *in the case of* t_1 *and between* $[0, 0.5]$ *in the case of* t_5*. Let us assume that both fire at the same speed with* $v_1 = v_5 = 0.5$*. The other transitions are weakly enabled and firing all of them with* 0.5 *is an admissible solution. Assume that these values are chosen. It is easy to see that this corresponds to a steady-state. The evolutions of the markings of* p_1*,* p_2 *and* p_3 *are illustrated in Figure 5.5.*

5.4 State estimation of timed contPN with relaxed finite server semantics

Through this section we assume that $T_o = \emptyset$, that is, no transition is observed, and we try to estimate the possible markings after some time has elapsed. This represents a time-reachability problem, in the sense that the reachability space will depend not only on the net structure $\mathcal{N}$ and the initial marking $\boldsymbol{m}_0$ but also on time. Let us define the following sets:

1. $RS_\tau(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} | \exists$ an admissible IFS vector $\boldsymbol{v}(\cdot) : \boldsymbol{m} = \boldsymbol{m}_0 + \int_0^\tau \boldsymbol{C} \cdot \boldsymbol{v}(\tau) \cdot d\tau\}$, that is the set of markings in which the net may be at time τ.
2. $RS^t(\mathcal{N}, \boldsymbol{m}_0) = \bigcup_{\tau \geq 0} RS_\tau(\mathcal{N}, \boldsymbol{m}_0)$ that represents the set of markings reachable in the timed system.

Example 5.13. *Let us consider the contPN system in Fig. 5.2 with* $w = 1$ *and assume* $\mathcal{V}(t_1) = [V_m^1, V_M^1] = [0,1]$ *and* $\mathcal{V}(t_2) = [V_m^2, V_M^2] = [0,1]$*. At time* $\tau = 0.1$*, the set of reachable markings*

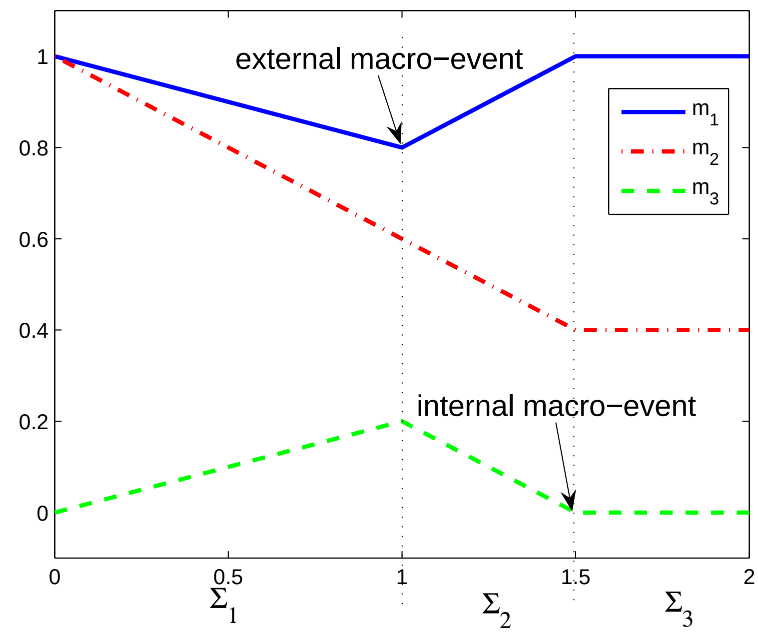


Figure 5.5: Marking evolution of a continuous system under relaxed finite server semantics

is: $RS_{0.1}(\mathcal{N}, \boldsymbol{m}_0) = \{[m_1, m_2]^T | m_1 \in [0.9, 1], m_2 \in [0, 0.1], m_1 + m_2 = 1\}$ *because the maximum number of tokens that can be removed from* p_1 *is* $V_M^1 \cdot \tau = 0.1$ *and the maximum number of tokens that can enter in* p_2 *is* $V_M^1 \cdot \tau = 0.1$*. At* $\tau = 0.2$*,* $RS_{0.2}(\mathcal{N}, \boldsymbol{m}_0) = \{[m_1, m_2]^T | m_1 \in [0.8, 1], m_2 \in [0, 0.2], m_1 + m_2 = 1\}$*. The reachability space of the timed system is:* $RS^t(\mathcal{N}, \boldsymbol{m}_0) = \{[m_1, m_2]^T | 0 \le m_1, m_2 \le 1, m_1 + m_2 = 1\} = RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$.

Note that we assume that the IFS vector is kept constant during a macro-period. As shown before, some markings are reachable in the limit in the untimed continuous system (see Ex. 2.7). In the case of (relaxed) finite server semantics, since the flow is kept constant these markings can be effectively reached in finite time.

Example 5.14. *Going back to the contPN in Fig. 5.2 but assuming now* $w = 2$*, it is obvious that the marking* $[0, 0]^T$ *is lim-reachable in the untimed model. While as timed, if* $\mathcal{V}(t_i) = [0, 1]$ *then* $\boldsymbol{v} = [1, 1]^T$ *is an admissible firing speed and* $[0, 0]^T$ *is reached after 1 time unit.*

If the minimum firing speed of each transition is "0" then all the markings that are lim-reachable in the untimed net are reachable in the timed one.

Theorem 5.15. *Let* $\langle \mathcal{N}, \boldsymbol{m}_0, \mathcal{V} \rangle$ *be a timed contPN and* $\forall t_j \in T$*,* $V_m^j = 0$*. Then* $lim\text{−}RS^{ut}(\mathcal{N}, \boldsymbol{m}_0) = RS^t(\mathcal{N}, \boldsymbol{m}_0)$.

Proof. Obviously, $RS^t(\mathcal{N}, \boldsymbol{m}_0) \subseteq lim - RS^{ut}(\mathcal{N}, \boldsymbol{m}_0)$. In fact each marking $\boldsymbol{m}$ that is reachable in a timed net satisfies the state equation and, since we are assuming that empty cycles cannot be fired, according to Theorem 2.10 the same firing sequence also ensures that $\boldsymbol{m}$ is also lim-reachable in the untimed net.

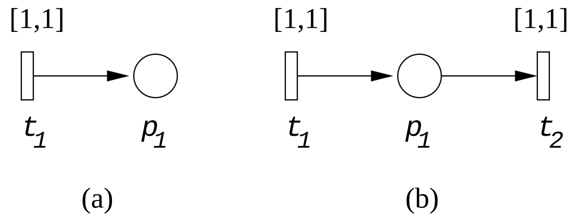


Figure 5.6: ContPN systems in which some markings reachable as untimed cannot be reached in the timed model.

Conversely, let us take $\boldsymbol{m} \in lim-RS^{ut}(\mathcal{N},\boldsymbol{m}_0)$, therefore, according to Theorem 2.10, exists a vector $\boldsymbol{\sigma}$ such that $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma}$ and a firing sequence σ with the same support that is fireable at $\boldsymbol{m_0}$. Hence transitions in the support of $\boldsymbol{\sigma}$ cannot belong to empty cycles.

Let us construct a IFS $\boldsymbol{v}$ using $\boldsymbol{\sigma}$ that can be fired in the timed net. First, let $V_M^{min} = \min\limits_{\forall j,\sigma_j>0}\{V_M^j\}$ be the maximum firing speed at which a proportion of $\boldsymbol{\sigma}$ can fire and $\sigma^{max} = \max\limits_j\{\boldsymbol{\sigma}_j\}$. Now, $\boldsymbol{v} = \frac{V_M^{min}}{\sigma^{max}}\cdot\boldsymbol{\sigma}$ can be fired in the timed net since for every $v_j = \frac{V_M^{min}}{\sigma^{max}}\cdot\sigma_j$ the following is true: $0 \le V_M^{min}\cdot\frac{\sigma_j}{\sigma^{max}} \le V_M^{min} \le V_M^j$. If $\boldsymbol{v}$ is fired for a time $\frac{\sigma^{max}}{V_M^{min}}$ then $\boldsymbol{m}$ is reached in the timed model. □

In the previous theorem, the condition that the minimum firing speed of every transitions is zero is fundamental. If it is not satisfied there can exist markings that are lim-reachable in the untimed system but not reachable in the timed one. This happens because with a minimum firing speed greater than zero, some transition firing sequences are not possible in the timed system.

Example 5.16. *Let us go back to the timed contPN system of Fig. 5.2 with $w = 2$ and let us assume now $\mathcal{V}(t_1) = [0.1,0.1]$ and $\mathcal{V}(t_2) = [0.1,0.1]$. In the untimed system, $\boldsymbol{m} = [0,0.5]^T$ is reachable firing $\sigma = t_1$ but in the timed net system it is not since $v_1(\tau) = v_2(\tau) = 0.1, \forall\tau$ implying $\dot{m}_2(\tau) = v_1(\tau) - v_2(\tau) = 0$ with $m_2(0) = 0$. Hence, place p_2 remains empty.*

The reachability space of a timed contPN system is, by definition, the union of all markings that can be reached in a time $\tau \ge 0$. In general, the reachability space is not a monotonous function of time, i.e, given two time instants $\tau_1 \le \tau_2$, the condition $RS_{\tau_1}(\mathcal{N},\boldsymbol{m}_0) \subseteq RS_{\tau_2}(\mathcal{N},\boldsymbol{m}_0)$ does not necessarily hold.

Example 5.17. *Let us consider the timed contPN in Fig. 5.6(a). For $\tau_0 = 0$, $RS_0(\mathcal{N},\boldsymbol{m}_0) = \{[0]\}$ but for $\tau_1 = 1$, $RS_1(\mathcal{N},\boldsymbol{m}_0) = \{[1]\}$ because transition t_1 has $v_1(\tau) = 1, \forall\tau > 0$.*

However, under some conditions this monotonicity property holds.

Theorem 5.18. *Let $\langle\mathcal{N},\boldsymbol{m}_0,\mathcal{V}\rangle$ be a timed contPN and $\forall t_j \in T$, $V_m^j = 0$. If $\tau_1 \le \tau_2$ then $RS_{\tau_1}(\mathcal{N},\boldsymbol{m}_0) \subseteq RS_{\tau_2}(\mathcal{N},\boldsymbol{m}_0)$.*

Proof. Since the minimum firing speed of each transition is null then all the markings that are reachable in a time τ_1 can be reached in τ_2 just stopping all transitions after τ_1. □

Computation of the reachability space of a timed contPN system is very difficult as long as it is necessary to compute the markings reached in a time τ for all $\tau \geq 0$. In the case of contPN system that it is bounded as timed there exists a time instant τ_{min} such that $\bigcup_{0\leq\tau\leq\tau_{min}} RS_\tau(\mathcal{N},\boldsymbol{m}_0) = RS^t(\mathcal{N},\boldsymbol{m}_0)$. Moreover, if $V_m^j = 0$ for all $t_j \in T$, according to Theorem 5.18 $\bigcup_{0\leq\tau\leq\tau_{min}} RS_\tau(\mathcal{N},\boldsymbol{m}_0) = RS_{\tau_{min}}(\mathcal{N},\boldsymbol{m}_0)$ and $RS^t(\mathcal{N},\boldsymbol{m}_0) = RS_{\tau_{min}}(\mathcal{N},\boldsymbol{m}_0)$. In other words, the markings reached before τ_{min} form the reachability space of the timed net system.

Proposition 5.19. *Let $\langle\mathcal{N},\boldsymbol{m}_0,\mathcal{V}\rangle$ be a timed contPN and $\forall t_j \in T, \mathcal{V}(t_j) = [0, V_M^j]$. There exists τ_{min} such that $RS_\tau(\mathcal{N},\boldsymbol{m}_0) = RS^t(\mathcal{N},\boldsymbol{m}_0)$, $\forall\tau \geq \tau_{min}$ iff the net is bounded as timed.*

Proof. "$\Longrightarrow$" Let us assume that the net is not bounded as timed. Then exists a place p_i whose marking is growing firing at least one transition t_j. If m_i is reached in minimum τ_0 time units, then the infinite sequence $m_i, m_i+1, m_i+2, m_i+3, \ldots$ is reached at (minimum) time instants $\tau_0 < \tau_1 < \tau_2 < \tau_3 < \ldots$. This is impossible because by hypothesis there exists τ_{min} such that all the markings can be reached in this time. Hence the net is bounded as timed.

"$\Longleftarrow$" If the net is bounded as timed the reachability space is a closed convex and each marking can be reached in the finite time, thus there exists a τ such that every marking can be reached in a time τ' with $\tau' \leq \tau$. The minimum firing speed is assumed to be null, then according to Theorem 5.18 all markings reachable in a time $\tau'' \geq \tau$ are reachable in a time τ. Taking $\tau_{min} = \tau$, the conclusion is satisfied. □

Observe that in the previous theorem we require only time boundedness not boundedness as untimed.

Example 5.20. *Let us consider the net in Fig. 5.6(b). This net is not bounded as untimed because t_1 can fire infinitely and the marking of p_1 is unbounded. But this net is timed bounded for the time intervals associated, and according to Prop. 5.19, there exists τ_{min} such that all reachable markings can be reached in a time inferior to τ_{min}. For this system, $\tau_{min} = 0$ because $RS^t(\mathcal{N},\boldsymbol{m}_0) = \{\boldsymbol{m}_0\}$.*

An interesting problem is the computation of such τ_{min} ensuring that each reachable marking is reachable within this time. Here we characterize τ_{min} for a particular class of nets (consistent and conservative) that although restricted, are significant for many real applications. The idea of these computations is to search for the longest time to reach the markings at the border of $lim-RS^{ut}$.

Definition 5.21. *Let $\langle\mathcal{N},\boldsymbol{m_0}\rangle$ be a contPN system. A marking $\boldsymbol{m}_1 \in lim-RS^{ut}$ is an* extreme marking *if it is not inside of any line segment contained in $lim-RS^{ut}$. In other words, if $\boldsymbol{m}_1 = \alpha\cdot\boldsymbol{m}_2 + (1-\alpha)\cdot\boldsymbol{m}_3$ where $\boldsymbol{m}_2,\boldsymbol{m}_3 \in lim-RS^{ut}$, implies $\alpha = 0$ or $\alpha = 1$, then $\boldsymbol{m}_1$ is an* extreme marking. ■

Proposition 5.22. *Let $\langle\mathcal{N},\boldsymbol{m_0}\rangle$ be a consistent, conservative contPN system, each transition can be fired at least once and $\boldsymbol{m}_1 \in lim-RS^{ut}(\mathcal{N},\boldsymbol{m}_0)$. If there exists a P-semiflow $\boldsymbol{y}$ such that $\forall p_i \in ||\boldsymbol{y}||$, $\boldsymbol{m}_1[p_i] \neq \max_{\boldsymbol{m}\in lim-RS^{ut}(\mathcal{N},\boldsymbol{m}_0)} \{\boldsymbol{m}[p_i]\}$ then $\boldsymbol{m}_1$ is not an extreme marking.*

Proof. Let $\boldsymbol{m}_1 \in lim-RS^{ut}(\mathcal{N},\boldsymbol{m}_0)$ and $\boldsymbol{y}$ a P-semiflow such that $\forall p_i \in ||\boldsymbol{y}||$, $\boldsymbol{m}_1[p_i] \neq \max\{\boldsymbol{m}[p_i]\}$. Since in every place support of $\boldsymbol{y}$, the marking is not maximal then $\exists p_k, p_l$ such that $\boldsymbol{m}_1[p_k], \boldsymbol{m}_1[p_l] > 0$ with $p_k, p_l \in ||\boldsymbol{y}||$. We construct two reachable markings such that $\boldsymbol{m}_1$ is the midpoint of the line segment defined by these markings. Using the fact that p_k and p_l are the support of the same P-semiflow and their corresponding markings at $\boldsymbol{m}_1$ are neither maximum, neither minimum, there exists $\alpha > 0$ such that $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ defined as:

$$\boldsymbol{m}_2[p_h] = \begin{cases} \boldsymbol{m}_1[p_h], \text{ if } p_h \neq p_k \text{ and } p_h \neq p_l \\ \boldsymbol{m}_1[p_h] + \alpha, \text{ if } p_h = p_k \\ \boldsymbol{m}_1[p_h] - \frac{\boldsymbol{y}_f[p_k]}{\boldsymbol{y}_f[p_l]} \cdot \alpha, \text{ if } p_h = p_l \end{cases}$$
$$\boldsymbol{m}_3[p_h] = \begin{cases} \boldsymbol{m}_1[p_h], \text{ if } p_h \neq p_k \text{ and } p_h \neq p_l \\ \boldsymbol{m}_1[p_h] - \alpha, \text{ if } p_h = p_k \\ \boldsymbol{m}_1[p_h] + \frac{\boldsymbol{y}_f[p_k]}{\boldsymbol{y}_f[p_l]} \cdot \alpha, \text{ if } p_h = p_l \end{cases}$$

are reachable according to Theorem 2.12. It is obvious that $\frac{1}{2}(\boldsymbol{m}_2 + \boldsymbol{m}_3) = \boldsymbol{m}_1$ and $\boldsymbol{m}_1 \neq \boldsymbol{m}_2 \neq \boldsymbol{m}_3$ and according to Def. 5.21, $\boldsymbol{m}_1$ is not an extreme marking. □

Using the previous proposition, the set of extreme markings can be computed for the class of conservative and consistent contPN just ensuring that in each P-semiflow there exists one place marked with the maximum number of tokens. Let us denote by $P_M \subseteq P$ a subset of places such that for every P-semiflow $\boldsymbol{y}_i$, $|\{||\boldsymbol{y}_i|| \cap P_M\}| = 1$. In other words, there exists only one place in P_M support of any P-semiflow $\boldsymbol{y}_i$. Considering $\boldsymbol{p}_M : P \to [0,1]$ such that $\boldsymbol{p}_M[p_i] = 1$ if $p_i \in P_M$ and $\boldsymbol{p}_M[p_i] = 0$ otherwise, let us present the following linear programming problem (LPP):

$$\begin{aligned} \min\ & \tau - M \cdot \boldsymbol{p}_M^T \cdot \boldsymbol{m} \\ \text{s.t.} & \begin{cases} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{h} \\ \tau \cdot \boldsymbol{V}_m \leq \boldsymbol{h} \leq \tau \cdot \boldsymbol{V}_M \end{cases} \end{aligned} \tag{5.5}$$

where, M is a big value such that the performance index corresponds to the minimum time τ to reach the maximum number of tokens in places P_M; $\boldsymbol{h} = \boldsymbol{v} \cdot \tau$ and it is introduced to obtain a linear state equation; the last constraints are the bounds for the IFS written in terms of $\boldsymbol{h}$; $\boldsymbol{V}_m$ and $\boldsymbol{V}_M$ are the vectors containing the minimum and the maximum for IFS.

Theorem 5.23. *Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a consistent, conservative contPN system, each transition can be fired at least once and $\forall\, t_j \in T$, $V_m^j = 0$. For any $\tau \geq \tau_{min}$ where $\tau_{min} = \max \tau_k$ with τ_k the solutions of LPP* (5.5) *for all possible sets P_M, $RS_\tau(\mathcal{N}, \boldsymbol{m}_0) = RS^t(\mathcal{N}, \boldsymbol{m}_0)$.*

Proof. According to Theorem 5.18, all markings reachable in a time $\tau < \tau_{min}$ can be reached in a time τ_{min}. We have to prove that all markings in $RS^t(\mathcal{N}, \boldsymbol{m}_0)$ can be reached in the time τ_{min}. Since τ_{min} is the minimum time to reach all extreme markings, it is enough to prove that all other markings at the border of the reachability space can be reached in τ_{min}. Obviously, the interior points of the reachability space are reached in a time less than the time to reach the markings at the borders.

Let $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ be two extreme markings. We are going to prove that $\boldsymbol{m}_1$, a linear combination of these two markings can be reached in a time equal to the minimum time needs to reach $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$. Since $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ are reachable, there exist $0 \leq \boldsymbol{v}_2 \leq \boldsymbol{V}_M$, τ_2, $0 \leq \boldsymbol{v}_3 \leq \boldsymbol{V}_M$ and τ_3, such that $\boldsymbol{m}_2 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{v}_2 \cdot \tau_2$ and $\boldsymbol{m}_3 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{v}_3 \cdot \tau_3$.

Computing $\boldsymbol{m}_1 = \alpha \cdot \boldsymbol{m}_2 + (1 - \alpha) \cdot \boldsymbol{m}_3$ from the previous equations, we obtain: $\boldsymbol{m}_1 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot (\alpha \cdot \boldsymbol{v}_2 \cdot \tau_2 + (1 - \alpha) \cdot \boldsymbol{v}_3 \cdot \tau_3)$. Let us assume $\tau_2 \leq \tau_3$, then $\boldsymbol{m}_1$ can be reached first obtaining an intermediate marking: $\boldsymbol{m}'_1 = \boldsymbol{m}_0 + \boldsymbol{C}\cdot(\alpha\cdot\boldsymbol{v}_2\cdot\tau_2 + (1-\alpha)\cdot\boldsymbol{v}_3\cdot\tau_2)$ and then $\boldsymbol{m}_1 = \boldsymbol{m}'_1 + \boldsymbol{C}\cdot(1-\alpha)\cdot\boldsymbol{v}_3\cdot(\tau_3-\tau_2)$. The marking $\boldsymbol{m}_1$ is reachable from $\boldsymbol{m}'_1$ because the conditions of Theorem 2.12 are satisfied.

The time need to reach $\boldsymbol{m}_1$ is $\tau' = \alpha\cdot\tau_2 + (1-\alpha)\cdot(\tau_3-\tau_2) = (2\cdot\alpha-1)\cdot\tau_2 + (1-\alpha)\cdot\tau_3 \leq (2\cdot\alpha-1)\cdot\tau_3 + (1-\alpha)\cdot\tau_3 \leq \alpha\cdot\tau_3 \leq \alpha\cdot\tau_3 \leq \tau_3$. □

Example 5.24. *Let us consider the timed contPN system in Fig. 5.3 with* $\mathcal{V}(t_1) = \mathcal{V}(t_2) = \mathcal{V}(t_4) = \mathcal{V}(t_5) = [0,1]$, $\mathcal{V}(t_3) = \mathcal{V}(t_6) = [0,0.1]$. *This net has one P-semiflow:* $\boldsymbol{y} = [5,5,5,2]^T$. *Solving LPP* (5.5) *for* $V_M = \{p_i\}$, $i = 1,\ldots,4$ *we obtain the following results: for* p_1 *the minimum time to reach* $\boldsymbol{m} = [2.8,0,0,0]^T$ *is* 20 *t.u., for* p_2 *the minimum time to reach* $\boldsymbol{m} = [0,2.8,0,0]^T$ *is* 20 *t.u., for* p_3 *the minimum time to reach* $\boldsymbol{m} = [0,0,2.8,0]^T$ *is* 20 *t.u., for* p_4*, the minimum time to reach* $\boldsymbol{m} = [0,0,0,7]^T$ *is* 50 *t.u. corresponding to the firing of* $\boldsymbol{h} = [3.5;3.5;0;0;0;5]^T$. *Hence for* $\tau \geq 50$ *all lim-reachable markings of the untimed model can be reached in the timed one.*

The computation of such τ_{min} is important because for the state estimation without any measurement, and if $\boldsymbol{V}_m = \boldsymbol{0}$, if the time is greater than τ_{min} then all reachable markings are possible. If the time at which the estimation is performed is less than τ_{min}, the following constraints provide the space of all possible markings, that, in fact, is the set $RS_\tau(\mathcal{N},\boldsymbol{m}_0)$:

$$\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{h}(\tau) \tag{5.6}$$

$$\tau\cdot\boldsymbol{V}_m \leq \boldsymbol{h}(\tau) \leq \tau\cdot\boldsymbol{V}_M \tag{5.7}$$

Obviously, for each marking, the corresponding vector $\boldsymbol{h}$ should be such that there is no empty cycle that fires. In the case of conservative and consistent contPN with all transitions fireable and $\boldsymbol{V}_m = \boldsymbol{0}$, if the time that is considered is greater than τ_{min} then the constraint (5.7) can be ignored and the possible states belong to $RS^t(\mathcal{N},\boldsymbol{m}_0)$.

5.5 Going back to finite server semantics

In this chapter we have considered relaxed finite server semantics in which the main difference with respect to the original definition is that we are letting the firing speed of a transition belong to an interval and not be limited to its maximal value. In this part we are going to consider controlled finite server semantics, showing how the previous results can be used to prove some controllability aspects. We can define a control as an external command that can slow down the firing speed of a transition and in some cases stop the flow. In the next chapter we will motivate the choice of this control. Here let us see that this choice is consistent with the discrete case where a controllable transition can be fired or not if it is enabled but cannot fire if it is not enabled. Therefore the flow of a transition under controlled finite server semantics, $\boldsymbol{w}(\tau)$, will be given by:

$$\begin{cases} \boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau) \\ \boldsymbol{b}(\tau) \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau) \end{cases} \tag{5.8}$$

where $\boldsymbol{f}$ is the flow of an autonomous contPN system with finite server semantics given by (2.7) and $\boldsymbol{b} \geq \boldsymbol{0}$ is a vector that specifies the minimum firing speed of the transitions. We assume here that all transitions are controllable, i.e., admit an external control.

It is immediate to observe that according to Definition 5.11 taking $\boldsymbol{b} = \boldsymbol{V}_m$ contPN with finite server semantics with control is equivalent to relaxed finite server semantics.

Almost all results regarding the reachability of the relaxed model presented in the previous section are proved in the case in which the minimal firing speed of each transition is zero, i.e., $\boldsymbol{V}_m = \boldsymbol{b} = \boldsymbol{0}$. Considering this assumption, the flow of a controlled contPN system with finite server semantics is given by the following set of equations:

$$\begin{cases} \boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau) \\ \boldsymbol{0} \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau) \end{cases} \tag{5.9}$$

where the flow is kept constant during a macro-period, i.e., the flow changes either when the controller change the nominal value of at least one transition or when a place becomes empty.

A first question is which are the markings that can be reached in the controlled finite server semantics. A direct application of Theorem 5.15 is:

Corollary 5.25. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a contPN system with finite server semantics with all transitions controllable. All markings lim-reachable in the untimed contPN system are reachable in the controlled finite server semantics system.*

Definition 5.26. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a contPN system with finite server semantics. $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is controllable if for all $\boldsymbol{m}_f \in RS^t$, $\boldsymbol{m}_f$ can be reached in* finite time.

According to Prop. 5.19, the characterization of the controllability arises:

Corollary 5.27. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a contPN system with finite server semantics with all transitions controllable. $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is controllable iff the net is timed bounded.*

Proof. According to Prop. 5.19 there exists a finite τ_{min} such that all markings can be reached at this time iff the net system is timed bounded. Therefore, the net system is controllable iff the net system is timed bounded. □

The computation of τ_{min} such that all markings can be reached in the timed model for the class of conservative and consistent nets is based on some LPP 5.5. This value ensures that for a time greater than τ_{min} there exists for sure a control law such that all markings can be reached, i.e., it is the minimum time to reach all possible markings.

Corollary 5.28. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a conservative and consistent contPN with finite server semantics with all transitions fireable and controllable. The minimum time to reach all markings in the reachability space is $\tau_{min} = \max \tau_k$ with τ_k the solutions of LPP* (5.5) *for all possible sets P_M.*

Example 5.29. *Let us consider the conservative and consistent contPN in Fig. 5.3. Let us assume finite server semantics with $\boldsymbol{\lambda} = [1, 1, 0.1, 1, 1, 0.1]^T$. Considering that all transition are controllable, this net is controllable since it is bounded. According to Ex. 5.24, $\tau_{min} = 50$ t.u. is the minimum time to reach all markings. Observe that $\boldsymbol{m} = [0, 0, 0, 7]^T$ is reached in a time equal to* 50 *t.u. with the firing vector $\boldsymbol{f} = [3.5; 3.5; 0; 0; 0; 5]^T$ hence this bound is reached.*

Chapter 6

On controllability and steady state control

Summary

This chapter addresses several questions related to the control of timed continuous Petri Nets under infinite server semantics. First, some results regarding equilibrium states and control actions are given. In particular, it is shown that the considered systems are piecewise linear, and for every linear subsystem the possible steady-states are characterized. Second, optimal steady-state control is studied, a problem that surprisingly can be computed in polynomial time, when all transitions are controllable and the objective function is linear. Third, an interpretation of some controllability aspects in the framework of linear dynamic systems is presented. An interesting finding is that non controllable poles are zero valued.

6.1 Introduction

In this chapter we will study some aspects regarding the control of contPN systems with infinite server semantics. We are showing how an external command, called control is introduced in the case of this systems and then we are studying some problems regarding the steady state control. In general, given a contPN it is impossible to say which will be the steady state, therefore, we will study all possible steady states, the equilibrium markings. For some subclasses, an unique solution exists. In other cases, many equilibrium points exist but all of them have the same flow, a desirable property in many situations.

To compute an optimal steady state, minimizing/maximizing a linear profit function, a LPP is purposed. In the case in which all transitions are controllable, this LPP provide an optimal solution. Otherwise, a branch & bound algorithmic extension can be used.

A bridge between controllability in *classical linear theory* and Petri nets is established. The simplifying idea is to keep the fact that the dynamic model is multilinear, but ignore the constraints that must be respected by the action: non-negative and upper bounded by a function of the marking (state). It is shown that net systems generate different token conservation laws, some of them leading to uncontrollability. Some conservation laws are generated by the P-flows (which depend only on the net structure) and zero valued poles appear in the *uncontrollable* part of the system. Other zero valued *controllable* poles are related to conservation laws that depend on the net structure, the firing rates and the token load of P-(semi)flows. Finally, some *controllable* non zero poles may generate token conservation laws for particular values of $\boldsymbol{m}_0$.

6.2 Controllability of linear systems

Let us consider a time-invariant linear system expressed by:

$$\begin{cases} \dot{\boldsymbol{x}}(\tau) = \boldsymbol{A} \cdot \boldsymbol{x}(\tau) + \boldsymbol{B} \cdot \boldsymbol{u}(\tau) \\ \boldsymbol{y}(\tau) = \boldsymbol{S} \cdot \boldsymbol{x}(\tau) + \boldsymbol{D} \cdot \boldsymbol{u}(\tau) \end{cases} \tag{6.1}$$

where $\boldsymbol{x}(\tau) \in \boldsymbol{X}^n$ is the state of the system, $\boldsymbol{u}(\tau) \in \boldsymbol{U}^m$ the input control and $\boldsymbol{y}(\tau) \in \boldsymbol{Y}^l$ is the output.

Definition 6.1. *[45][72] A dynamic system* (6.1) *is said to be* completely state controllable *if for any time τ_0 and any final state, it is possible to construct an* unconstrained *control vector $\boldsymbol{u}(\tau)$ that will transfer the initial state $\boldsymbol{x}(\tau_0)$ to the final state $\boldsymbol{x}(\tau)$ in a finite time.*

A very well-known controllability criterion exists which allows to decide whether a continuous linear system is controllable or not. Given a linear system (6.1), the *controllability matrix is* defined as:

$$\mathbb{C} = [\boldsymbol{B} \cdots \boldsymbol{A}^k \boldsymbol{B} \cdots \boldsymbol{A}^{(n-1)} \boldsymbol{B}] \tag{6.2}$$

Proposition 6.2. *[45][72] A linear continuous-time system* (6.1) *is completely controllable iff $\mathbb{C}$ is full rank (i.e. $rank(\mathbb{C}) = n$). If $\mathbb{C}$ is not a full rank matrix then the controllable subspace has dimension $rank(\mathbb{C})$.*

Equation (6.1) corresponds to a state-space representation of the system description. Other representation is the input/output one. Applying Laplace transform to the first equation in (6.1) and considering null initial conditions ($\boldsymbol{x}(0) = \boldsymbol{0}$, which can always be obtained by translation) we have: $s \cdot \boldsymbol{x}(s) = \boldsymbol{A} \cdot \boldsymbol{x}(s) + \boldsymbol{B} \cdot \boldsymbol{u}(s)$ and combining with the second equation, the *transfer-matrix function* is obtained:

$$\boldsymbol{G}(s) = \frac{\boldsymbol{y}(s)}{\boldsymbol{u}(s)} = \boldsymbol{S} \cdot (s \cdot \boldsymbol{I} - \boldsymbol{A})^{-1} \cdot \boldsymbol{B} + \boldsymbol{D} = \frac{\boldsymbol{S} \cdot adj(s \cdot \boldsymbol{I} - \boldsymbol{A})\boldsymbol{B} + \boldsymbol{\Delta}(s)\boldsymbol{D}}{\boldsymbol{\Delta}(s)} \tag{6.3}$$

where, $adj(s \cdot \boldsymbol{I} - \boldsymbol{A})$ is the adjoint of the matrix $(s \cdot \boldsymbol{I} - \boldsymbol{A})$ and $\boldsymbol{\Delta}(s)$ the determinant of the same matrix.

The roots of the denominator of the transfer function are called *the poles* of the system and they can be obtained by solving the characteristic equation: $\boldsymbol{\Delta}(s) = det(s \cdot \boldsymbol{I} - \boldsymbol{A}) = 0$. Notice that the poles of the transfer function matrix and the eigenvalues of the matrix $\boldsymbol{A}$ are the same.

The poles play a very important role in system analysis and design. For example, if all poles have negative real part then the system is stable (for any bounded input, the output is bounded). If one pole has positive real part then the system is unstable. Zero valued poles correspond to integrators.

6.3 Controllability of timed continuous Petri nets: problem statement

In section 3.2 it is shown that, in general, a contPN system under infinite server semantics is not monotone neither w.r.t. firing speed of the transitions $\boldsymbol{\lambda}$ nor w.r.t. the initial marking $\boldsymbol{m}_0$. Therefore, for the faster evolution of the system it is not necessary that all transitions work at their maximum firing speed. Since a transition is associated in general to a machine and this machine cannot work faster then its maximum firing rate, the only control action we can consider is to brake it down.

The parameters $\boldsymbol{\lambda}$ associated with the transitions in timed contPNs with infinite server semantics represent their *firing rate*. We assume that the only action that can be applied is to *reduce* their firing flow i.e. throughput. If a transition can be controlled (its flow reduced or even stopped), we will say that it is a *controllable* transition. The flow of a controlled transition t_i becomes $f_i - u_i$, where f_i is the flow of the *unforced* system (i.e. defined as in Eq. (2.9)) and u_i is the control action $0 \le u_i \le f_i$.

Definition 6.3. *The flow of the forced (or controlled) timed contPN is denoted as* $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$*, with* $0 \le \boldsymbol{u}(\tau) \le \boldsymbol{f}(\tau)$*, where* $\boldsymbol{u}(\tau)$ *represents the control input.*

According to the above notation, the controlled flow vector is $\boldsymbol{w} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} - \boldsymbol{u} \ge 0$, with $u_i = 0$ if t_i is not controllable. Thus the state equation of controlled timed contPNs (i.e. net systems in which all the transitions are controllable: $\forall t \in T$, $\boldsymbol{u}[t] > 0$ is possible at certain instant) becomes:

$$\begin{cases} \dot{\boldsymbol{m}} = \boldsymbol{C} \cdot (\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} - \boldsymbol{u}) \\ 0 \le \boldsymbol{u} \le \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} \end{cases} \tag{6.4}$$

This is a particular hybrid system: piecewise linear with autonomous switches and dynamic (or state-based) constraints in the input.

Example 6.4. *Let us consider the net system in Fig. 2.4 with* $\boldsymbol{\lambda} = \mathbf{1}$*. It is ruled by the following set of systems of the form* (6.4)*:*

- $\boldsymbol{m} \in R_1$: $\begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -1 & 1 & 1 & 0 \\ \frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{2} & 0 & -1 & 0 \\ -1 & -1 & 3 & 0 \end{bmatrix} \boldsymbol{m}(\tau) - \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ -2 & -1 & 3 \end{bmatrix} \boldsymbol{u}(\tau) \\ \boldsymbol{0} \leq \boldsymbol{u}(\tau) \leq \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{m}(\tau) \end{cases}$

- $\boldsymbol{m} \in R_2$: $\begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} -1 & 0 & 1 & 1 \\ \frac{1}{2} & 0 & 0 & -1 \\ \frac{1}{2} & 0 & -1 & 0 \\ -1 & 0 & 3 & -1 \end{bmatrix} \boldsymbol{m}(\tau) - \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ -2 & -1 & 3 \end{bmatrix} \boldsymbol{u}(\tau) \\ \boldsymbol{0} \leq \boldsymbol{u}(\tau) \leq \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{m}(\tau) \end{cases}$

- $\boldsymbol{m} \in R_3$: $\begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 0 & -1 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & \frac{1}{2} \\ 0 & -1 & 3 & -1 \end{bmatrix} \boldsymbol{m}(\tau) - \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ -2 & -1 & 3 \end{bmatrix} \boldsymbol{u}(\tau) \\ \boldsymbol{0} \leq \boldsymbol{u}(\tau) \leq \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{m}(\tau) \end{cases}$

- $\boldsymbol{m} \in R_4$: $\begin{cases} \dot{\boldsymbol{m}}(\tau) = \begin{bmatrix} 0 & \frac{1}{2} & 1 & -1 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & -1 & \frac{1}{2} \\ 0 & -\frac{1}{2} & 3 & -1 \end{bmatrix} \boldsymbol{m}(\tau) - \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ -2 & -1 & 3 \end{bmatrix} \boldsymbol{u}(\tau) \\ \boldsymbol{0} \leq \boldsymbol{u}(\tau) \leq \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{m}(\tau) \end{cases}$

■

Unless otherwise stated, *during this chapter we will assume that all transitions are controllable*, i.e., can be slowed down by an external controlling agent. It may also be possible to extend the approach to deal with uncontrollability of certain transitions. If transition t_j cannot be controlled, then it is obvious that the control input must be $u_j = 0$ at every time instant.

6.4 Control of timed contPNs and characterization of steady-states

Controlling all transitions, almost all reachable markings of an untimed system can be reached in the timed one. The only problem is at the borders when the marking of one place is zero. In this case, the marking is reached at the limit (this is like the discharging of a capacitor in an electrical RC-circuit: theoretical total discharging takes an infinite amount

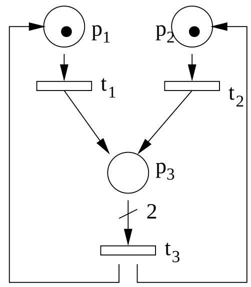


Figure 6.1: Timed continuous Join-Free system with $\boldsymbol{\lambda} = [1,1,1]^T$. It has a unique equilibrium point for a given $\boldsymbol{u}_d$ (for example $\boldsymbol{m}_d = [0.66, 0.66, 0.66]^T$ for $\boldsymbol{u}_d = [0,0,0]^T$).

of time). For example, in the net system in Figure 6.1 the marking $[0,1,1]^T$ is reachable in the untimed model. Considering now the timed model, stopping transitions t_2 and t_3 ($u_2 = f_2$ and $u_3 = f_3$) and setting $u_1 = 0$, the marking $[0,1,1]^T$ is reached at the limit because $\dot{m}_1(\tau) = -\lambda_1 \cdot m_1(\tau) \Rightarrow m_1(\tau) = e^{-\lambda_1 \cdot \tau} \cdot m_1(0)$. Note that it takes an infinite amount of time to empty p_1.

The steady-state markings we are interested *to obtain* (reference markings for the control loop) are strictly positive (if the marking of a place is zero then the flows of its output transitions are zero, meaning total inactivity of the machines or processors being controlled). These markings can be reached in finite time in the timed model. Let us first prove the following Lemma:

Lemma 6.5. *If all transitions are controllable, any fireable sequence in the untimed model that does not empty any place in the process, can be fired in the timed-controlled model in finite time.*

Proof. The same sequence can be reproduced, for example firing one transition each time and stopping the others. Since the firings do not empty any place, they take finite time. □

Proposition 6.6. *If all transitions are controllable:*

1) if $\boldsymbol{m}$ is reachable in the timed-controlled model then it is reachable in the untimed model (i.e., $RS^t \subseteq RS^{ut}$)

2) if $\boldsymbol{m}$ is reachable in the untimed model then it is lim-reachable in the timed-controlled model (i.e., $RS^{ut} \subseteq lim - RS^t$);

3) If $\boldsymbol{m} > 0$ is reachable in the untimed model, it can be reached in finite time in the timed-controlled model (i.e., $RS^{ut}+ \subseteq RS^t$).

Proof. 1) If $\boldsymbol{m}$ is reachable in the timed-controlled model then, according to (6.4), $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \int_0^\tau \boldsymbol{C} \cdot \boldsymbol{w}(\theta) \cdot d\theta = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \int_0^\tau \boldsymbol{w}(\theta) \cdot d\theta$. Let $\boldsymbol{\sigma} = \int_0^\tau \boldsymbol{w}(\theta) \cdot d\theta \geq 0$. Since $\boldsymbol{m}$ is reachable in the timed model no trap can be empty at $\boldsymbol{m}$ in $\mathcal{N}_{\boldsymbol{\sigma}}$ (a marked place cannot be emptied). Moreover, since $\boldsymbol{\sigma}$ is fireable in the timed model, a sequence with the same support can be fired in the untimed model. Hence $\boldsymbol{m}$ is reachable in the untimed net according to Prop. 2.9.

2) If $\boldsymbol{m}$ is reachable in the untimed model from $\boldsymbol{m}_0$, there exists a sequence $\sigma = \alpha_1 t_1 \alpha_2 t_2 \ldots \alpha_k t_k$ that leads from $\boldsymbol{m}_0$ to $\boldsymbol{m}$ passing throug the intermediary markings: $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, ..., $\boldsymbol{m}_{k-1}$. This sequence is equivalent to an infinite sequence $\sigma^1 \sigma^2 \ldots$ defined as:

$$\begin{array}{rcl} \sigma^i & = & (\beta_{i,1}\alpha_1)t_1(\beta_{i,2}\alpha_2)t_2\cdots(\beta_{i,k}\alpha_k)t_k \\ \beta_{1,j} & = & 1/2^j, (j=1,\ldots,k) \\ \beta_{i,1} & = & 1/2^i,\ (i=1,2,\ldots) \\ \beta_{i,j} & = & \frac{1}{2}\left(\sum_{l=1}^{i}\beta_{l,j-1}-\sum_{l=1}^{i-1}\beta_{l,j}\right), (i\geq 2\ j\geq 2). \end{array}$$

Intuitively, in the first round the proportion of firing is decreasing each time so that places are never emptied by more than one half. In the following rounds, it is taken into account how much the previous transitions in the sequence have been fired, and how much the actual transition has been fired until now, again to be sure that the reduction never exceeds one half.

Formally, consider an intermediate step in which $\sigma^1\ldots\sigma^{i-1}$ and only part of σ^i, namely,

$$(\beta_{i,1}\alpha_1)t_1(\beta_{i,2}\alpha_2)t_2\ldots(\beta_{i,j-1}\alpha_{j-1})t_{j-1},$$

have been fired. If we denote

$$\boldsymbol{c}_j=\alpha_j\boldsymbol{C}(\cdot,t_j)$$

the actual marking can be described as

$$\begin{array}{rcl} \boldsymbol{m}_{i,j-1} & = & \boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{c}_1+\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}\right)\boldsymbol{c}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{c}_j+\ldots+\left(\sum_{h=1}^{i-1}\beta_{k,j}\right)\boldsymbol{c}_k= \\ & = & \left(1-\left(\sum_{h=1}^{i}\beta_{h,1}\right)\right)\boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}\right)(\boldsymbol{m}_0+\boldsymbol{c}_1)+\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}\right)\boldsymbol{c}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{c}_j+\ldots+\left(\sum_{h=1}^{i-1}\beta_{k,j}\right)\boldsymbol{c}_k= \\ & = & \left(1-\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{m}_1+ \\ & & +\left(\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{c}_2+\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}\right)\boldsymbol{c}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{c}_j+\ldots+\left(\sum_{h=1}^{i-1}\beta_{k,j}\right)\boldsymbol{c}_k= \\ & = & \left(1-\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}-\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{m}_1+\sum_{h=1}^{i}\beta_{h,2}(\boldsymbol{m}_1+\boldsymbol{c}_2) \\ & & +\left(\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{c}_2+\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}\right)\boldsymbol{c}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{c}_j+\ldots+\left(\sum_{h=1}^{i-1}\beta_{k,j}\right)\boldsymbol{c}_k=\ldots \\ & = & \left(1-\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}-\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{m}_1+\sum_{h=1}^{i}\beta_{h,2}\boldsymbol{m}_2 \\ & & +\left(\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{c}_2+\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}\right)\boldsymbol{c}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{c}_j+\ldots+\left(\sum_{h=1}^{i-1}\beta_{k,j}\right)\boldsymbol{c}_k=\ldots \\ & = & \left(1-\sum_{h=1}^{i}\beta_{h,1}\right)\boldsymbol{m}_0+\left(\sum_{h=1}^{i}\beta_{h,1}-\sum_{h=1}^{i}\beta_{h,2}\right)\boldsymbol{m}_1+ \\ & & +\ldots+\left(\sum_{h=1}^{i}\beta_{h,j-1}-\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{m}_{j-1}+ \\ & & +\left(\sum_{h=1}^{i-1}\beta_{h,j}-\sum_{h=1}^{i-1}\beta_{h,j-1}\right)\boldsymbol{m}_j+\ldots+ \\ & & +\left(\sum_{h=1}^{i}\beta_{h,k-1}-\sum_{h=1}^{i-1}\beta_{h,k}\right)\boldsymbol{m}_{k-1}+ \\ & & \left(\sum_{h=1}^{i}\beta_{h,k}\right)\boldsymbol{m}_k \end{array}$$

Hence,

$$\boldsymbol{m}_{i,j-1}\geq\left(\sum_{h=1}^{i}\beta_{h,j-1}-\sum_{h=1}^{i-1}\beta_{h,j}\right)\boldsymbol{m}_{j-1}$$

and so t_{r_j} can be fired half of this amount and no place looses more that one half of its token content. Therefore, each σ^i can be fired in the timed-controlled model (Lemma 6.5).

With respect to the convergence to σ, it can be proved that

$$\beta_{i,j} = \frac{(i+j-2)!}{(j-1)!(i-1)!} \cdot \frac{1}{2^{i+j-1}},$$

which is the probability mass distribution of the negative binomial of parameters j, $1/2$. Applying induction, the proof is based on the fact that the cumulative distribution function F_j can be immediately expressed as a regularized incomplete beta function, i.e., $F_j(h) = I_{1/2}(j, h+1)$, and that a regularized incomplete beta function enjoys the following property:

$$I_{1/2}(a,b) - I_{1/2}(a+1,b) = \frac{(a+b-1)!}{(a)!(b-1)!} \cdot \frac{1}{2^{a+b}}.$$

Observe that

$$\begin{aligned} \beta_{1,j} &= \frac{1}{2^j} = \frac{(1+j-2)!}{(j-1)!(1-1)!} \cdot \frac{1}{2^{1+j-1}}, \\ \beta_{i,1} &= \frac{1}{2^i} = \frac{(i+1-2)!}{(1-1)!(i-1)!} \cdot \frac{1}{2^{i+1-1}}. \end{aligned}$$

Applying induction "following the rows", assume it holds for $\beta_{l,k}$, with $1 \le l \le i-1$ and $1 \le k \le n$, and for $\beta_{i,k}$, with $1 \le k \le j-1$. Let us prove it for $\beta_{i,j}$:

$$\begin{aligned} \beta_{i,j} &= \frac{\sum_{l=1}^{i} \beta_{i,j-1} - \sum_{l=1}^{i-1} \beta_{i,j}}{2} = \frac{\beta_{i,j-1}}{2} + \frac{\sum_{l=1}^{i-1} \beta_{i,j-1} - \sum_{l=1}^{i-1} \beta_{i,j}}{2} \\ &= \frac{\beta_{i,j-1}}{2} + \frac{1}{2}\left(\sum_{l=1}^{i-1} \frac{\binom{l+j-3}{j-2}}{2^{l+j-2}} - \sum_{l=1}^{i-1} \frac{\binom{l+j-2}{j-1}}{2^{l+j-1}} \right) \\ &= \frac{\beta_{i,j-1}}{2} + \frac{1}{2}\left(\sum_{l=0}^{i-2} \frac{\binom{l+j-2}{j-2}}{2^{l+j-1}} - \sum_{l=0}^{i-2} \frac{\binom{l+j-1}{j-1}}{2^{l+j}} \right) \\ &= \frac{1}{2} \frac{\binom{i+j-3}{j-2}}{2^{i+j-2}} + \frac{1}{2}\left(I_{1/2}(j-1, i-1) - I_{1/2}(j, i-1)\right) \\ &= \frac{1}{2^{i+j-1}} \frac{(i+j-3)!}{(j-2)!(i-1)!} + \frac{1}{2^{i+j-1}} \frac{(i+j-3)!}{(j-1)!(i-2)!} \\ &= \frac{1}{2^{i+j-1}} \frac{(i+j-2)!}{(j-1)!(i-1)!} \end{aligned}$$

This means that the amount in which transition t_j is fired is α_j times a cumulative distribution function, and so in the limit it converges to α_j.

Observe that $\boldsymbol{m}$ is reached at the limit, being an infinite sequence.

3) Let $\boldsymbol{m} > 0$ be such that a sequence $\sigma = \alpha_1 t_1 \cdots \alpha_i t_i \cdots \alpha_j t_j \cdots \alpha_k t_k$ exists that leads from $\boldsymbol{m}_0$ to $\boldsymbol{m}$ in the untimed model. If the firing of σ does not empty any place, Lemma 6.5 can be applied. Otherwise, we prove that a control law exists that brings the timed model from

$\boldsymbol{m}_0$ to a marking $\boldsymbol{m}' = \boldsymbol{m}_0 + \boldsymbol{C}\boldsymbol{\sigma}' > 0$ such that $\boldsymbol{m}$ can be reached from $\boldsymbol{m}'$ using other control law.

To construct σ', let us assume without loss of generality that when firing t_i and t_j while firing σ, at least one place in ${}^\bullet t_i$ and one place in ${}^\bullet t_j$ become empty. Define $\sigma' = \alpha_1 t_1 \ldots \ldots \frac{1}{2}\alpha_i t_i \ldots \frac{1}{4}\alpha_j t_j \ldots \frac{1}{4}\alpha_k t_k$. This sequence can be fired in the timed-controlled model, since the amounts in which t_i and t_j are fired ensure that no place is emptied.

The desired marking is reachable from $\boldsymbol{m}'$ according to Prop. 2.9: $\boldsymbol{m} = \boldsymbol{m}' + \boldsymbol{C} \cdot \boldsymbol{\sigma}''$ with $\boldsymbol{\sigma}'' = \boldsymbol{\sigma} - \boldsymbol{\sigma}' \geq 0$, $\boldsymbol{m}' > 0$ implying the existence of a firing sequence with the same support as $\boldsymbol{\sigma}''$ since all the transitions of the net system are fireable; and $\boldsymbol{m} > 0$ means that no empty trap exists at $\boldsymbol{m}$.

Now, the control law to go from $\boldsymbol{m}'$ to $\boldsymbol{m}$ is constructed. Since $\boldsymbol{m} > 0$ and $\boldsymbol{m}' > 0$, the sequence $\sigma'' = \sigma - \sigma' = \frac{1}{2}\alpha_i t_i \ldots \frac{3}{4}\alpha_j t_j \ldots \frac{3}{4}\alpha_k t_k$ can be reordered in such a way that no place is emptied. For example, imagine σ'' starts firing $\frac{1}{2}\alpha_i t_i$. This would empty a place $p_h \in {}^\bullet t_i$. However, since $\boldsymbol{m}[p_h] > 0$, at least one transition puts more tokens in p_h than the amount removed by the firing of $\frac{1}{2}\alpha_i t_i$. Let t_k be one of those transitions such that the marking of p_h is positive after firing it, $\sigma'' = \frac{1}{2}\alpha_i t_i \beta_r t_r \ldots \beta_k t_k \ldots$.

Let γ be the number of tokens in p_h after firing $\beta_k t_k$, and define $\epsilon < \frac{\gamma}{\alpha_i + \gamma}$. Clearly the sequence $\frac{1}{2}\alpha_i (1-2\epsilon) t_i \beta_r (1-\epsilon) t_r \ldots \beta_k (1-\epsilon) t_k$ can be fired without emptying any place. Now, $\boldsymbol{m}[p_h] = \boldsymbol{Post}[p_h, t_k](1-\epsilon)\beta_k - \boldsymbol{Pre}[p_h, t_i](1-\epsilon)\alpha_i = (1-\epsilon)(\boldsymbol{Post}[p_h, t_k]\beta_k - \boldsymbol{Pre}[p_h, t_i]\alpha_i) = (1-\epsilon)\gamma > \epsilon\alpha_i$. Hence, $\epsilon\alpha_i t_i \epsilon\beta_r t_r \ldots \epsilon\beta_k t_k$ can be fired now.

Clearly this procedure can be extended to the case in which several places are emptied. Constructing a new firing sequence not emptying any place, Lemma 6.5 can be applied. □

Definition 6.7. *Let* $\mathbf{0} \leq \boldsymbol{u}_d \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}_d) \cdot \boldsymbol{m}_d$. *Then* $\boldsymbol{m}_d \in lim - RS^t$ *is an equilibrium point for* $\boldsymbol{u}_d$ *if* $\boldsymbol{C} \cdot (\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}_d) \cdot \boldsymbol{m}_d - \boldsymbol{u}_d) = \mathbf{0}$.

An equilibrium point represents a state in which the system can be maintained using the defined control action. Given $\boldsymbol{m}_0$ (initial) and $\boldsymbol{m}_d$ (desired) markings, one control problem is to reach $\boldsymbol{m}_d$ and then keep it. In this section we concentrate on the properties of steady states.

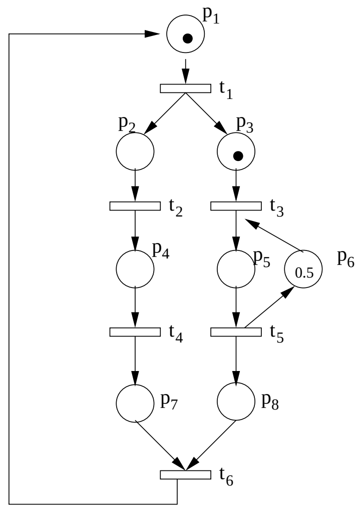
Obviously, taking into account (6.4), $\boldsymbol{m}_d \in RS^t$ is a equilibrium marking if together with the control input $\boldsymbol{u}_d$ is a solution of the following system:

$$\begin{aligned} &\boldsymbol{C} \cdot (\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}_d) \cdot \boldsymbol{m}_d - \boldsymbol{u}_d) = \mathbf{0} \\ &\mathbf{0} \leq \boldsymbol{u}_d \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}_d) \cdot \boldsymbol{m}_d \end{aligned} \tag{6.5}$$

Therefore, the steady-state flow of a controlled timed contPN $\boldsymbol{w} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}_d) \cdot \boldsymbol{m}_d - \boldsymbol{u}_d$ is a T-semiflow of the net. Notice that if the net is not consistent, some transitions will be stopped in steady-state, i.e., $\boldsymbol{w}$ will contain some zero components.

Given a $\boldsymbol{u}_d$, let us denote as $\boldsymbol{M}_{\boldsymbol{u}_d}$ all the equilibrium states it could maintain. That is, $\boldsymbol{M}_{\boldsymbol{u}_d} = \{\boldsymbol{m} \in lim - RS^t | \boldsymbol{C} \cdot (\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} - \boldsymbol{u}_d) = 0 \text{ and } 0 \leq \boldsymbol{u}_d \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m}\}$. The set $\boldsymbol{M}_{\boldsymbol{u}_d}$ can have one single element (Figure 6.1) or an infinite number of equilibrium markings in a single configuration ($\{(p_1, t_1), (p_2, t_2), (p_3, t_3), (p_4, t_4), (p_5, t_5), (p_7, t_6)\}$ in Figures 6.2 and 6.3), or infinite equilibrium markings in several configurations ($\{(p_1, t_1), (p_4, t_2), (p_7, t_3), (p_5, t_4), (p_6, t_5), (p_8, t_6)\}$ and $\{(p_1, t_1), (p_4, t_2), (p_7, t_3), (p_5, t_4), (p_6, t_5), (p_9, t_6)\}$ in Figures 6.4 and 6.5).

Next proposition characterizes all the equilibrium points of a net system with the same control action in steady state, $\boldsymbol{u}_d$.



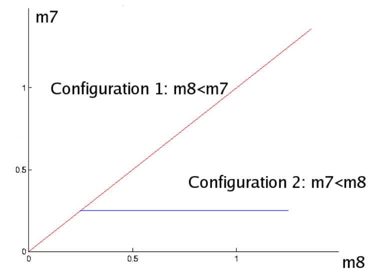
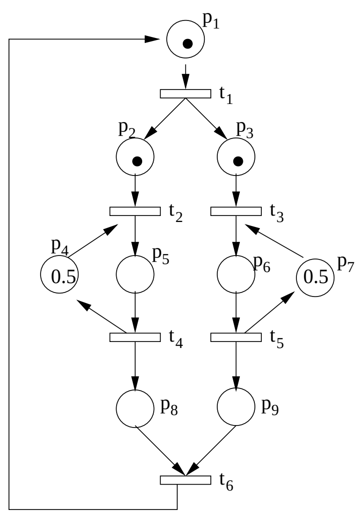


Figure 6.2: Timed continuous Marked Graph system with $\boldsymbol{\lambda} = [1,1,1,1,1,1]^T$ and many equilibrium points in the same configuration for a given $\boldsymbol{u}_d$.

Figure 6.3: Equilibrium points of the timed continuous Marked Graph system in Fig. 6.2 for $\boldsymbol{u}_d = [0,0,0,0,0,0]^T$.



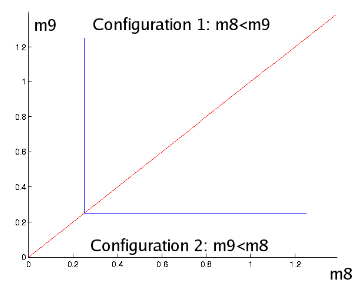


Figure 6.4: Timed continuous Marked Graph system with $\boldsymbol{\lambda} = [1,1,1,1,1,1]^T$ and many equilibrium points in several configurations for a given $\boldsymbol{u}_d$.

Figure 6.5: Equilibrium points of the timed continuous Marked Graph system in Fig. 6.4 for $\boldsymbol{u}_d = [0,0,0,0,0,0]^T$.

Proposition 6.8. *Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a consistent contPN system with all transitions fireable at least once. Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be the timed contPN system and $\boldsymbol{m}_d$ an equilibrium point for $\boldsymbol{u}_d$. Then $\boldsymbol{m}_i \geq 0$ is also an equilibrium point (reachable in finite time if $\boldsymbol{m}_i > 0$) for $\boldsymbol{u}_d$ iff:*

$$\begin{cases} \boldsymbol{B_y}^T \cdot (\boldsymbol{m}_d - \boldsymbol{m}_i) = 0 & (a) \\ \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot (\boldsymbol{\Pi}_d \cdot \boldsymbol{m}_d - \boldsymbol{\Pi}_i \cdot \boldsymbol{m}_i) = 0 & (b) \\ 0 \leq \boldsymbol{u}_d \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}_i \cdot \boldsymbol{m}_i & (c) \end{cases} \tag{6.6}$$

with $\boldsymbol{B}_y$ a basis of P-flows.

Proof. $\Longrightarrow$ If $\boldsymbol{m}_i$ is an equilibrium point then it is a reachable marking. The system is consistent so: $\boldsymbol{B}_y^T \cdot \boldsymbol{m}_i = \boldsymbol{B}_y^T \cdot \boldsymbol{m}_d$, i.e. (6.6.a) is necessary.

Both markings are equilibrium points: $\boldsymbol{C}\cdot(\boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}_d\cdot\boldsymbol{m}_d-\boldsymbol{u}_d) = 0$ and $\boldsymbol{C}\cdot(\boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}_i\cdot\boldsymbol{m}_i-\boldsymbol{u}_d) = 0$. Subtracting $\boldsymbol{u}_d$ from both equations, (6.6.b) is obtained.

$\Longleftarrow$ Equation (6.6.a) ensures the reachability of $\boldsymbol{m}_i$ according to Prop. 2.12. The control input $\boldsymbol{u}_d$ can be applied (6.6.c), and using (6.6.b) $\boldsymbol{m}_i$ is an equilibrium marking. □

Lemma 6.9. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0\rangle$ be a timed contPN system and $\boldsymbol{m}_d$, $\boldsymbol{m}_i$ two equilibrium points for $\boldsymbol{u}_d$. The flows at these markings are equal iff $\boldsymbol{\Pi}_d \cdot \boldsymbol{m}_d = \boldsymbol{\Pi}_i \cdot \boldsymbol{m}_i$.*

Proof. Flows are equal: iff $\boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}_d\cdot\boldsymbol{m}_d-\boldsymbol{u}_d = \boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}_i\cdot\boldsymbol{m}_i-\boldsymbol{u}_d$, that is iff $\boldsymbol{\Lambda}\cdot(\boldsymbol{\Pi}_d\cdot\boldsymbol{m}_d-\boldsymbol{\Pi}_i\cdot\boldsymbol{m}_i) = 0$. Since $\boldsymbol{\Lambda}$ is a full rank matrix (by definition is a diagonal matrix with diagonal elements greater than zero), this can happen iff $\boldsymbol{\Pi}_d \cdot \boldsymbol{m}_d = \boldsymbol{\Pi}_i \cdot \boldsymbol{m}_i$. □

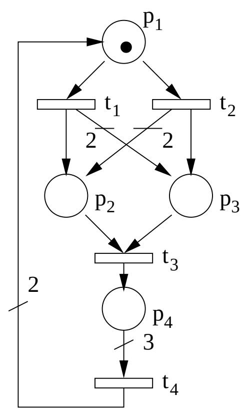


Figure 6.6: Conservative but not lim-live continuous EQ system with several equilibrium points for $\boldsymbol{\lambda} = [1,1,1,1]^T$, with different flow.

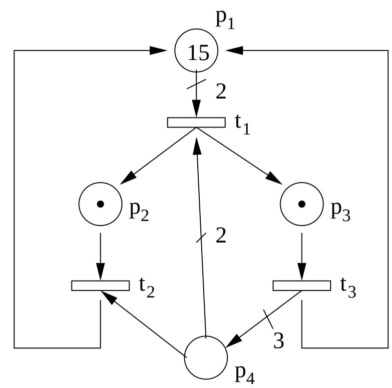


Figure 6.7: Bounded and lim-live contPN that has several equilibrium points with distinct flow.

Example 6.10. *For the timed contPN system depicted in Figure 6.6 the optimal flow $\boldsymbol{w}_{max} = [0.2,0.2,0.6,0.2]^T$ is obtained with $\boldsymbol{u}_d = [0,0,0,0]^T$ and marking $\boldsymbol{m}_d = [0.2,0.6,0.6,0.6]^T$. Marking $\boldsymbol{m}' = [0.1,0.3,1.8,0.3]^T$, is also an equilibrium point for $\boldsymbol{u}_d = [0,0,0,0]^T$, and the flow is different $\boldsymbol{w}' = [0.1,0.1,0.3,0.1]^T$. Obviously, the conditions of the lemma do not hold, $\boldsymbol{\Pi}_d \cdot \boldsymbol{m}_d \neq \boldsymbol{\Pi}' \cdot \boldsymbol{m}'$.*

Let $\boldsymbol{B}_x$ be a basis of T-flows of a net (i.e. $\boldsymbol{C}\cdot\boldsymbol{B}_x = \boldsymbol{0}$).

Theorem 6.11. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0\rangle$ be a consistent timed contPN system with all transitions fireable at least once. In one configuration $\boldsymbol{\Pi}$ all the equilibrium points for a given $\boldsymbol{u}$ have the same flow if*

$$rank\begin{bmatrix} \boldsymbol{\Lambda}\cdot\boldsymbol{\Pi} & | & \boldsymbol{B}_x \\ \boldsymbol{B}_y^T & | & 0 \end{bmatrix} = rank\begin{bmatrix} \boldsymbol{\Pi} \\ \boldsymbol{B}_y^T \end{bmatrix} + |T| - rank(\boldsymbol{C})$$

Proof. Let us assume that $\boldsymbol{m_a}$ and $\boldsymbol{m_b}$ are two equilibrium points under $\boldsymbol{\Pi}$ for the same control $\boldsymbol{u}$. Obviously, the flow in steady state will be a T-semiflow: $\boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}\cdot\boldsymbol{m_a} - \boldsymbol{u} = \boldsymbol{B}_x\cdot\boldsymbol{\alpha}$ ($\boldsymbol{B}_x\cdot\boldsymbol{\alpha} =$

$\sum_i \alpha_i \cdot \boldsymbol{b_{x_i}}$), and $\boldsymbol{\Lambda}\cdot\boldsymbol{\Pi}\cdot\boldsymbol{m_b} - \boldsymbol{u} = \boldsymbol{B}_x\cdot\boldsymbol{\beta}$. Now, subtracting both equations: $\boldsymbol{\Pi}\cdot\Delta\boldsymbol{m} - \boldsymbol{\Lambda}^{-1}\cdot\boldsymbol{B}_x\cdot\boldsymbol{\zeta} = 0$ ($\Delta\boldsymbol{m} = \boldsymbol{m_a} - \boldsymbol{m_b}$, $\boldsymbol{\zeta} = \boldsymbol{\alpha} - \boldsymbol{\beta}$). Moreover, since these markings are reachable, $\boldsymbol{B}_y^T\cdot\Delta\boldsymbol{m} = 0$.

$$\begin{bmatrix} \boldsymbol{\Pi} & | & -\boldsymbol{\Lambda}^{-1}\cdot\boldsymbol{B}_x \\ \boldsymbol{B}_y^T & | & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{m} \\ \boldsymbol{\zeta} \end{bmatrix} = 0 \tag{6.7}$$

Under the rank condition, the vectorial spaces generated by the row vectors of $[\boldsymbol{\Pi}^T|\boldsymbol{B}_y]$ and $[(\boldsymbol{\Lambda}^{-1}\boldsymbol{B}_x)^T|0]$ are linearly independent. Hence the null element is the only vector that belongs to both of them, i.e., $\boldsymbol{\Lambda}^{-1}\boldsymbol{B}_x\cdot\boldsymbol{\zeta} = 0$. Moreover, $\boldsymbol{\Lambda}^{-1}$ is a diagonal matrix and the columns in $\boldsymbol{B}_x$ are linearly independent since they are a T-flows basis, and so $\boldsymbol{\zeta} = 0$. Therefore $\boldsymbol{m_a}$ and $\boldsymbol{m_b}$ have the same flow. □

Example 6.12. *Let us consider the contPN system in Figure 6.7 with $\boldsymbol{\lambda} = [2,1,1]^T$. The configuration $\{(p_4,t_1),(p_4,t_2),(p_3,t_3)\}$ with associated matrix $\boldsymbol{\Pi}$ can have several equilibrium points with different flows because the conditions of Theorem 6.11 are not satisfied. For this system,* $\boldsymbol{\Pi} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $\boldsymbol{B}_y^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 4 & 1 \end{bmatrix}$, $\boldsymbol{\Lambda}^{-1}\cdot\boldsymbol{B}_x = \begin{bmatrix} \frac{1}{2} \\ 1 \\ 1 \end{bmatrix}$ *and* $rank\begin{bmatrix} \boldsymbol{\Pi} & -\boldsymbol{\Lambda}\cdot\boldsymbol{B}_x \\ \boldsymbol{B}_y^T & 0 \end{bmatrix} = rank\begin{bmatrix} \boldsymbol{\Pi} \\ \boldsymbol{B}_y^T \end{bmatrix} = 4$. *If $\boldsymbol{u} = [0,0,0]^T$, the equilibrium markings $\boldsymbol{m}_1 = [15.25,1,0.75,0.75]^T$ and $\boldsymbol{m}_2 = [15.5,0.8,0.7,0.7]^T$ belonging to this configuration have the flows $\boldsymbol{w_1} = [0.75,0.75,0.75]^T$ and $\boldsymbol{w_2} = [0.7,0.7,0.7]^T$ respectively. Thus, any intermediate value is also possible.*

For the class of Equal Conflict contPN, if the conflicting transitions are not controlled (otherwise the visit ratio is changed by the control), we can prove that all equilibrium points in a configuration have the same flow under the same control. Obviously, the result can be extended considering that the conflict transitions are controlled but in such way that the visit ratio is not changed.

Theorem 6.13. *Let $\langle\mathcal{N},\boldsymbol{\lambda},\boldsymbol{m_0}\rangle$ be a bounded and lim-live EQ timed contPN system. Given $\boldsymbol{u}_d$ in which transitions in conflict are not controlled, there exists at least one equilibrium point. If there are more than one, all of them have the same flow.*

Proof. The throughput in steady state for unforced ($\boldsymbol{u}_d = 0$) continuous EQ nets can be computed using a linear programming problem [42]. More precisely, the throughput is obtained looking for the slowest P-semiflow. The solution is unique with respect to the flow, but there can exist more than one marking that respect the P-semiflows and have the same associated flow.

Assume $^\bullet t = p$, i.e., t is a non synchronizing transition, and $\boldsymbol{u}_d[t] \neq 0$. If the steady-state marking of p is $\boldsymbol{m}[p]$, we can reduce the value of $\boldsymbol{m}[p]$ to $\boldsymbol{m}'[p] = \boldsymbol{m}[p] - \frac{\boldsymbol{Pre}[p,t]}{\boldsymbol{\lambda}[t]}\cdot\boldsymbol{u}_d[t]$ corresponding to the same steady state flow. This will be: $\boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{m}'[p]}{\boldsymbol{Pre}[p,t]} = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p,t]} - \boldsymbol{u}_d[t]$, the same as in the original system (with $\boldsymbol{u}_d[t] \neq 0$). For every controlled transition we can apply the same technique (in the case of synchronizations we remove tokens from all input places) obtaining an equivalent system with $\boldsymbol{u}_d = \boldsymbol{0}$. For this system all the equilibrium points have the same flow. □

Figure 6.1, 6.2 and 6.4 are CF (thus EQ). Therefore, this theorem ensures that all their equilibrium points have the same flow for any constant control input $\boldsymbol{u}_d$. The following theorem provides a sufficient condition to guarantee that the equilibrium point of a configuration is unique.

Theorem 6.14. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a bounded and lim-live EQ timed contPN system. If*

$$rank \begin{bmatrix} \boldsymbol{\Pi}_d \\ \boldsymbol{B}_y{}^T \end{bmatrix} = |P|$$

and conflict transitions are not controlled, then at most one equilibrium marking exists under $\boldsymbol{\Pi}_d$ for a given $\boldsymbol{u}_d$.

Proof. Let $rank \begin{bmatrix} \boldsymbol{\Pi}_d \\ \boldsymbol{B}_y{}^T \end{bmatrix} = |P|$ and $\boldsymbol{m}_d$, $\boldsymbol{m}_i$ ($\boldsymbol{m}_i = \boldsymbol{m}_d + \Delta\boldsymbol{m}$) two equilibrium points under $\boldsymbol{\Pi}_d$ for $\boldsymbol{u}_d$. Using Theorem (6.13) all equilibrium points with the same action $\boldsymbol{u}_d$ have the same flow, i.e., $\boldsymbol{\Pi}_d \cdot \boldsymbol{m}_d = \boldsymbol{\Pi}_d \cdot \boldsymbol{m}_i$ or $\boldsymbol{\Pi}_d \cdot \Delta\boldsymbol{m} = 0$. Moreover, $\boldsymbol{B}_y^T \cdot \boldsymbol{m}_i = \boldsymbol{B}_y^T \cdot \boldsymbol{m}_d$, or $\boldsymbol{B}_y^T \cdot \Delta\boldsymbol{m} = 0$.

Under the rank assumption, the previous system has only one solution, $\Delta\boldsymbol{m} = 0$. So $\boldsymbol{m}_d = \boldsymbol{m}_i$. Hence, $\boldsymbol{\Pi}_d$ has at most one equilibrium point. ☐

Example 6.15. *Let us consider the net in Figure 6.4 and let*

$$\boldsymbol{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

define one configuration. One P-flow basis is:

$$\boldsymbol{B}_y^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Since $rank \begin{bmatrix} \boldsymbol{\Pi}_d \\ \boldsymbol{B}_y{}^T \end{bmatrix} = 8 < 9$ *(the number of places) this configuration may have an infinite number of equilibrium points. In particular (Fig. 6.5), for* $\boldsymbol{m}_0 = [1,1,1,0.5,0,0,0.5,0,0]$ *the configuration 1* $(\{(p_1,t_1),(p_4,t_2),(p_7,t_3),(p_5,t_4),(p_6,t_5),(p_8,t_6)\})$ *has an infinite number of equilibrium points.*

But, for $\boldsymbol{m}_0 = [1,0,0,0.5,0,0,0.5,0,0]$ *the contPN system has only one equilibrium marking. Thus, the condition in Theorem 6.14 is sufficient, but not necessary.*

Corollary 6.16. *Let $\mathcal{N}$ be a conservative and consistent JF contPN. Given $\boldsymbol{u}_d$ and assuming that the conflict transitions are not controlled, for any $\boldsymbol{m}_0$ only one equilibrium point exists in $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$.*

Proof. Since the net is JF, all the conflicts are CEQ (if $t_1, t_2 \in p^\bullet, {}^\bullet t_1 = {}^\bullet t_2 = p$), thus the net can be mapped into CF [42]. A CF net is conservative iff it is strongly-connected implying $rank(\boldsymbol{C}) = |T| - 1 = |SEQS| - 1$ [75]. A conservative and consistent contPN with $rank(\boldsymbol{C}) = |SEQS| - 1$ is structurally lim-live and bounded [64] and Theorem 6.14 can be applied. ☐

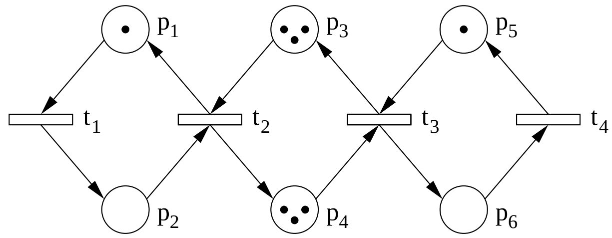


Figure 6.8: Timed contPN (marked graph) with several equilibrium points.

6.5 Optimal control for steady state

In production control, the profit function frequently depends on production sales, work in process (WIP) and amortization of investments. Under linear hypothesis for fixed machines (i.e. $\boldsymbol{\lambda}$ defined), the profit function may have the following form: $\boldsymbol{h}^T \cdot \boldsymbol{w} - \boldsymbol{z}^T \cdot \boldsymbol{m} - \boldsymbol{q}^T \cdot \boldsymbol{m}_0$, where $\boldsymbol{w}$ is the throughput vector, $\boldsymbol{h}^T$ a price vector w.r.t. flows, $\boldsymbol{m}$ the average marking, $\boldsymbol{z}^T$ is the WIP cost vector and $\boldsymbol{q}^T$ represents depreciations or amortization of the initial investments (over $\boldsymbol{m}_0$).

Let us consider the following linear programming problem:

$$\begin{array}{lll} \max & \boldsymbol{h}^T \cdot \boldsymbol{w} - \boldsymbol{z}^T \cdot \boldsymbol{m} - \boldsymbol{q}^T \cdot \boldsymbol{m}_0 & \\ s.t. & \boldsymbol{C} \cdot \boldsymbol{w} = 0,\ \boldsymbol{w} \geq 0 & \text{(a)} \\ & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \boldsymbol{m}, \boldsymbol{\sigma} \geq 0 & \text{(b)} \\ & w_i = \lambda_i \cdot \left(\frac{m_j}{\boldsymbol{Pre}[p_j, t_i]}\right) - \boldsymbol{v}[p_j, t_i], \forall t_i \in T, \forall p_j \in {}^{\bullet}t_i & \text{(c)} \end{array} \qquad (6.8)$$

where $\boldsymbol{v}[p_j, t_i]$ are *slack* variables.

The equations correspond to: (a) $\boldsymbol{w}$ is a T-semiflow; (b) fundamental equation ($\boldsymbol{m}$ is a reachable marking); (c) firing law for infinite server semantics.

Theorem 6.17. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed contPN system and let $\langle \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{v} \rangle$ be a solution of LPP (6.8), then*

1. *For every transition t_i, let $u_i = \min_{p_j \in {}^{\bullet}t_i} \boldsymbol{v}[p_j, t_i]$ be its control input. Then $\boldsymbol{u}$ is the control in steady-state for $\boldsymbol{m}$. (In the case of $|{}^{\bullet}t_i| = 1$ the corresponding slack variable is the same as the control input.)*

2. *If for every $u_i > 0$ transition t_i is controllable, then $\boldsymbol{u}$ is an optimal steady-state control.*

Proof. In steady state, $w_i = \lambda_i \cdot \min_{p_j \in {}^{\bullet}t_i} \left(\frac{m_j}{\boldsymbol{Pre}[p_j, t_i]}\right) - u_i$. Choosing $u_i = \min_{p_j \in {}^{\bullet}t_i} \boldsymbol{v}[p_j, t_i]$ for all transitions, the equation (6.8.c) is verified. If all t_i with $u_i \neq 0$ can be controlled, the control can be applied in steady state; then the command is optimal. □

For mono T-semiflow nets (conservative and consistent that have a unique minimal T-semiflow) (or nets reducible to mono T-semiflow [42]), the equation (6.8a) can be replaced with the equivalent one: $\boldsymbol{w} = \alpha \cdot \boldsymbol{x}$ (6.8a') with $\boldsymbol{x}$ the minimal T-semiflow.

If the net is consistent and every transition can be fired at least once, the equation (6.8b) is equivalent to: $\boldsymbol{B}_y^T \cdot \boldsymbol{m} = \boldsymbol{B}_y^T \cdot \boldsymbol{m}_0$, $\boldsymbol{m} \geq 0$ (6.8b').

Example 6.18. *The solution of LPP 6.8 is not necessarily unique (as we mentioned in the previous section). Let us see which is the maximum throughput in steady-state for the contPN in*

Figure 6.8 with $\boldsymbol{\lambda} = [1,1,1,1]^T$ *and* $\boldsymbol{m}_0 = [1,0,3,3,1,0]^T$. *Notice that this is a marked graph net system, hence is monotone and the optimal control should be* $\boldsymbol{u}_d = 0$. *Indeed, LPP (6.8) with (6.8a') and (6.8b') leads to:*

$$
\begin{array}{ll}
\max & w_1 \\
s.t. & w_1 = w_2 = w_3 = w_4 \qquad \textit{(6.8a')} \\
& \left.\begin{array}{l} m_1 + m_2 = m_5 + m_6 = 1 \\ m_3 + m_4 = 6 \end{array}\right\} \textit{(6.8b')} \\
& w_1 = m_1 - u_1 \\
& w_2 = m_2 - v_{22} \\
& w_2 = m_3 - v_{23} \\
& w_3 = m_4 - v_{34} \\
& w_3 = m_5 - v_{35} \\
& w_4 = m_6 - u_4 \\
& u_1, u_4 \geq 0 \\
& \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{v} \geq 0
\end{array}
\tag{6.9}
$$

One optimal solution of this LPP is: $w_1 = 0.5$, $\boldsymbol{m}_d = [0.5\ 0.5\ 3.5\ 2.5\ 0.5\ 0.5]^T$ *and* $\boldsymbol{v} = [0,0,3,2,0,0]^T$. *Therefore* $u_2 = \min(v_{22}, v_{23}) = 0$, $u_3 = \min(v_{34}, v_{35}) = 0$ *and* $\boldsymbol{u}_d = [0\ 0\ 0\ 0]^T$ *is an optimal control in steady state (*$\boldsymbol{u}_d = \boldsymbol{0}$ *leads always to optimal flow in marked graphs).*

For sure the solution is not unique: all the markings that satisfy (6.10) are also solution of (6.9).

$$
\left\{\begin{array}{l}
m_1 = m_2 = m_5 = m_6 = 0.5 \\
m_3 + m_4 = 6 \\
m_3, m_4 \geq 0.5
\end{array}\right.
\tag{6.10}
$$

Up to now we have considered that all transitions are controllable. What happens when some are uncontrollable? In the extreme case, in which *all* transitions are uncontrollable (the *unforced* system), the problem to compute the optimal steady-state (maximum throughput) was addressed in [42] and can be solved using a *branch and bound* algorithm. Let us assume $T = T_C \cup T_N$, where T_C is the set of controllable transitions and T_N the set of the uncontrollable transitions.

When all synchronizations are controllable ($\{t \text{ s.t. } |^\bullet t| > 1\} \subseteq T_C$), the problem remains polynomial time. In fact, it is the same problem as (6.8) in which slack variables are not allowed for the uncontrolled transitions, i.e., $\boldsymbol{v}[p_j, t_i] = 0$ for all t_i such that $|^\bullet t_i| = 1$. In this case, the solution of (6.8) can be reached with a specific control.

When a synchronization is not controllable, the problem may be more difficult. The corresponding slack variable cannot be used. As in [42] we can relax (6.8) and the flow of non controllable transitions will be upper bounded with inequalities written for every input place:

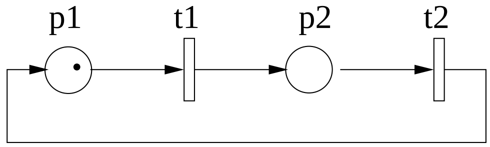


Figure 6.9: Join-Free timed contPN system with $\boldsymbol{\lambda} = [1,1]^T$, $t_1 \in T_N$, $t_2 \in T_C$.

$$
\begin{array}{lll}
\max & \boldsymbol{h}^T \cdot \boldsymbol{w} - \boldsymbol{z}^T \cdot \boldsymbol{m} - \boldsymbol{q}^T \cdot \boldsymbol{m}_0 & \\
s.t. & \boldsymbol{C} \cdot \boldsymbol{w} = 0,\ \boldsymbol{w} \geq 0 & \text{(a)} \\
 & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \boldsymbol{m}, \boldsymbol{\sigma} \geq 0 & \text{(b)} \\
 & w_i = \lambda_i \cdot \left(\frac{m_j}{\boldsymbol{Pre}[p_j, t_i]} \right) - \boldsymbol{v}[p_j, t_i], & \\
 & \forall p_j \in {}^\bullet t_i,\ t_i \in T_C,\ \boldsymbol{v}[p_j, t_i] \geq 0 & \text{(c)} \\
 & w_i = \lambda_i \cdot \left(\frac{m[p]}{\boldsymbol{Pre}[p, t_i]} \right),\ \text{if } p = {}^\bullet t_i,\ t_i \in T_N, & \text{(d)} \\
 & w_i \leq \lambda_i \cdot \left(\frac{m_j}{\boldsymbol{Pre}[p_j, t_i]} \right), \forall p_j \in {}^\bullet t_i,\ t_i \in T_N. & \text{(e)}
\end{array}
\tag{6.11}
$$

Because of (6.11.e), the LPP (6.11) provides in general a non tight bound, i.e. the solution may be non reachable. This occurs when none of the input places of a non controllable join transition really restricts the flow of that transition. Similar to [42], a branch and bound algorithm can be used. For every non controllable join transition t_j, a number of $|{}^\bullet t_j|$ LPPs should be computed by adding an equation that relates the flow of t_j with the marking of each one of its input places. Thus, a very similar algorithm with the one presented in Section 2.3.4, can be used in this situation.

6.6 Approaching dynamic control: on controllability and marking invariance laws

6.6.1 Definition of controllability

Controllability with constrained inputs

Assume the systems under study are described by the equations in (6.4). The classical control theory for linear systems cannot be applied because we are working inside a polytope (not in a *vectorial space*) and our control input is non negative and dynamically bounded.

Definition 6.19. *Given $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ and a set of controlled transitions $T_C \subset T$, a marking $\boldsymbol{m}_f$ is said to be a* lim-reachable steady-state *when there exists a constrained control action $\boldsymbol{u}(\tau)$ on T_C that is able to drive the marking from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ (in finite or infinite time) and maintain it.*

Definition 6.20. *The timed contPN $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ with a set of controlled transitions $T_C \subset T$ is* controllable *if $\forall \boldsymbol{m}_0$ and $\forall \boldsymbol{m}_f \geq 0$ such that $\boldsymbol{B_y^T} \cdot \boldsymbol{m}_0 = \boldsymbol{B_y^T} \cdot \boldsymbol{m}_f$, $\boldsymbol{m}_f$ is a lim-reachable steady-state.*

Unfortunately, the controllability of all transitions is required in order to obtain a controllable contPN system.

Example 6.21. *Let us consider the net system in Figure 6.9 and assume that only t_2 is controllable. The marking $\boldsymbol{m} = [1,0]^T$ cannot be an equilibrium marking because in steady-state $w_1 = w_2$ so $1 = 0 - \boldsymbol{u}[t_2]$ which implies a negative command on t_2. Therefore, $\boldsymbol{m}$ cannot be maintained in the timed and controlled model. In fact, any marking $\boldsymbol{m}'$ with $\boldsymbol{m}'[p_1] > \boldsymbol{m}'[p_2]$ is non maintainable, because $\dot{\boldsymbol{m}}'[p_1] = -\boldsymbol{m}'[p_1] + \boldsymbol{m}'[p_2] - \boldsymbol{u}[t_2] \le -\boldsymbol{m}'[p_1] + \boldsymbol{m}'[p_2] < 0$. Hence, the timed JF model of Figure 6.9 is not controllable with $T_C = \{t_2\}$.*

Proposition 6.22. *A pure and conservative timed contPN $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is controllable iff all transitions are controllable i.e. $T = T_C$.*

Proof. The sufficient condition is immediate: if all transitions allow a control action, the net is controllable. We can reach any desired marking (maybe at the limit) (Prop. 6.6) and then we stop all transitions (i.e. $\boldsymbol{u} = \boldsymbol{f}$).

Necessity: Let $\boldsymbol{m}_0$ be an initial marking that puts tokens in all the P-semiflows and let us assume t_i is not controllable. We are going to prove that a marking $\boldsymbol{m}$ satisfying $\boldsymbol{B_y^T} \cdot \boldsymbol{m}_0 = \boldsymbol{B_y^T} \cdot \boldsymbol{m}$ exists that cannot be maintained.

Let

$$\beta_i = \max\left\{ b | \boldsymbol{m} \ge b \cdot \boldsymbol{Pre}[\cdot, t_i] \text{ and } \boldsymbol{B_y^T} \cdot \boldsymbol{m}_0 = \boldsymbol{B_y^T} \cdot \boldsymbol{m} \right\} \tag{6.12}$$

In fact, β_i represents the enabling bound of t_i [16]. Let $\boldsymbol{m}$ be a solution of (6.12). Since β_i is obtained through maximization, and the P-semiflows are all marked, for sure $\boldsymbol{m}[p_j] > 0, \forall p_j \in {}^\bullet t_i$.

Since the net is pure and conservative, $t_i^\bullet \cap (T \setminus {}^\bullet t_i) \neq \emptyset$, then at least one place in $t_i^\bullet$ must be empty (otherwise the enabling degree could have been greater). Clearly this place cannot remain empty if t_i is not controlled. □

Classical approach: controllability without constrained inputs

During the rest of the chapter, a relaxation of the equations modeling the system is proposed, eliminating the restrictions related to the bounds of the control input. Therefore, the system under study is relaxed to the non-linear dynamical equation of (6.4):

$$\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} - \boldsymbol{C} \cdot \boldsymbol{u} \tag{6.13}$$

The goal is to understand better the behavior of contPN and interpret classical results in the contPN setting. In many cases, the regulation of the system is done to a point (desired marking + desired input) that is not at the boundary. In this case, a region around it can be defined in which the constraints are not active. Hence, this study is interesting when the system is considered in a local sense around these non boundary points.

Basically, the number of null eigenvalues are explored, eigenvalues that introduce token conservation laws. It will be seen that some of these conservation laws are given by the net structure $\mathcal{N}$ (the P-flows, Subsection 6.6.2), others depend on $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ (Subsection 6.6.3) and others depend also on the particular marking $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ (Subsection 6.6.4).

For classical linear systems *controllability* has been thoroughly studied. For contPN systems, every $\boldsymbol{\Pi}(\boldsymbol{m})$ leads to a linear and time-invariant dynamic system with controllability matrix $\mathbb{C}(\boldsymbol{m})$:

$$\mathbb{C}(\boldsymbol{m}) = -\left[\begin{array}{ccc} \boldsymbol{C} & \cdots & (\boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}))^{n-1} \cdot \boldsymbol{C} \end{array} \right] \tag{6.14}$$

Proposition 6.23. *If all transitions are controllable,* $\forall \boldsymbol{m} \in lim - RS^t$*, the spaces generated by the columns of* $\mathbb{C}(\boldsymbol{m})$ *and* $\boldsymbol{C}$ *are equal. Thus* $rank(\mathbb{C}(\boldsymbol{m})) = rank(\boldsymbol{C}) = |P| - \dim(\boldsymbol{B}_y)$.

Proof. Since the columns of $\boldsymbol{C}$ are contained in $\mathbb{C}$, it is immediate that the space generated by the columns of $\mathbb{C}$ contains the space generated by the columns of $\boldsymbol{C}$. Thus we only need to prove that it cannot be greater. Observe that $(\boldsymbol{C} \cdot \boldsymbol{\Lambda\Pi}(\boldsymbol{m}))^{n-1} \cdot \boldsymbol{C} = \boldsymbol{C} \cdot (\boldsymbol{\Lambda\Pi}(\boldsymbol{m}) \cdot \boldsymbol{C})^{n-1}$.

Thus, $\mathbb{C} = \boldsymbol{C} \cdot [\boldsymbol{I} \cdots (\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{C})^{n-1}]$. Notice that any P-flow of $\boldsymbol{C}$ is also a P-flow of $\mathbb{C}$. Hence, $rank(\mathbb{C}) = rank(\boldsymbol{C}) = |P| - \dim(\boldsymbol{B}_y)$. □

Notice that $\mathbb{C}(\boldsymbol{m})$ depends on $\boldsymbol{\Pi}(\boldsymbol{m})$, but the space generated by its columns is always the same, just that one defined by matrix $\boldsymbol{C}$. This is something that can be easily expected because all transitions have been assumed to be controllable.

6.6.2 Uncontrollable zero valued poles and decomposition

Token conservation laws given by the net structure (P-flows) produce non controllable contPN systems in a classical sense. This was observed in [57] and happens because the P-flows based token conservation laws make $|P| - rank(\boldsymbol{C})$ places linearly-redundant. Using a proper similitude transformation (the $\mathbb{Q}_{\mathcal{N}}$ matrix that will be given in Definition 6.25) it is possible to obtain a decomposition into a controllable subsystem and an uncontrollable one (similar to the Kalman controllable canonical form). The uncontrollable subsystem has only zero valued poles and they will be called *uncontrollable (zero) valued poles.*

Example 6.24. *Let us consider the contPN system in Figure 6.8 with* $\boldsymbol{\lambda} = [\alpha, \beta, \gamma, \delta]^T$*. This net has three linearly independent token conservation laws derived from P-(semi)flows:* $m_1 + m_2 = 1, m_3 + m_4 = 6$ *and* $m_5 + m_6 = 1$*. Thus* $\dot{m}_1 + \dot{m}_2 = \dot{m}_3 + \dot{m}_4 = \dot{m}_5 + \dot{m}_6 = 0$*, which means that three uncontrollable zero valued poles will appear.*

The following transformation matrix is used to change the reference in which the marking vector is expressed. This will be useful for studying the controllability of the system. The kind of transformation matrix to be considered will have in this context a particular structure.

Definition 6.25. *Let* $\mathcal{N}$ *be a contPN. A transformation matrix* $\mathbb{Q}_{\mathcal{N}}$*, is formed with rows from a basis of P-flows and elementary vectors in order to build a full rank matrix.*

Example 6.26. *For the timed models in Figure 6.9 and Figure 6.8, P-flow basis are respectively:*

$$\boldsymbol{B}_{y1}^T = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad \textit{and} \quad \boldsymbol{B}_{y2}^T = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{6.15}$$

Adding elementary vectors, $\mathbb{Q}$ *matrices can be, for example:*

$$\mathbb{Q}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \textit{and} \ \mathbb{Q}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.16}$$

The system described by equation (6.13) can be rewritten in new coordinates $\bar{\boldsymbol{m}}$, when matrix $\mathbb{Q}_{\mathcal{N}}$ is used as a state vector transformation matrix. Let $\bar{\boldsymbol{m}} = \mathbb{Q}_{\mathcal{N}} \cdot \boldsymbol{m}$.

Definition 6.27. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed contPN described by equation (6.13), where $\mathbb{Q}_{\mathcal{N}}$ is a transformation matrix of $\mathcal{N}$. Then*

$$\dot{\bar{\boldsymbol{m}}} = \mathbb{Q}_{\mathcal{N}} \boldsymbol{C} \boldsymbol{\Lambda} \boldsymbol{\Pi}(\boldsymbol{m}) \mathbb{Q}_{\mathcal{N}}^{-1} \bar{\boldsymbol{m}} - \mathbb{Q}_{\mathcal{N}} \boldsymbol{C} \boldsymbol{u} \tag{6.17}$$

will be called a $\mathbb{Q}$-canonical representation of equation (6.13).

Theorem 6.28. *Let $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a contPN system, then:*

1) In the linear system under $\boldsymbol{\Pi}(\boldsymbol{m})$ the number of zero valued poles is given by the dimension of the right annullers of $\boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m})$.

2) The number of non controllable poles is $|P| - rank(\boldsymbol{C})$ and they are zero valued.

Proof. 1) The zero eigenvalues of the matrix $\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\boldsymbol{m})$ are:

$$\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{v} = 0 \cdot \boldsymbol{v} = 0$$

2) Making the change of variables $\bar{\boldsymbol{m}} = \mathbb{Q}_{\mathcal{N}} \cdot \boldsymbol{m}$, (6.17) is obtained. For each P-flow of the basis one zero row appears in $\mathbb{Q}_{\mathcal{N}} \cdot \boldsymbol{C}$.

Without loss of generality, assume that the i^{th} row of $\mathbb{Q}_{\mathcal{N}} \cdot \boldsymbol{C}$ is zero. Then the i^{th} row of $\mathbb{Q}_{\mathcal{N}} \boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\boldsymbol{m})\mathbb{Q}_{\mathcal{N}}^{-1}$ is zero. Therefore the value of the state variable $\bar{\boldsymbol{m}}_i$ is never affected by other state variables, or by the input, thus $\bar{\boldsymbol{m}}_i$ is uncontrollable. Each one of these $\bar{\boldsymbol{m}}_i$ comes from a P-flow equation, a linear constraint among variables (i.e. token conservation law: $\boldsymbol{b_i}^T \cdot \bar{\boldsymbol{m}} = \boldsymbol{b_i}^T \cdot \bar{\boldsymbol{m}}_0$). Thus the pole value associated to $\bar{\boldsymbol{m}}_i$ is zero and there exist $\dim(\boldsymbol{B}_y)$ uncontrollable zero valued poles. According to Proposition 6.23, $rank(\mathbb{C}(\boldsymbol{m})) = |P| - \dim(\boldsymbol{B}_y)$, then there exist no more uncontrollable poles. Otherwise stated, if there are more zero valued poles, they are controllable (as we will see in Section 6.6.3). □

Example 6.29. *Let us consider the contPN system in Figure 6.9. It has the following equation:*

$$\dot{\boldsymbol{m}} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \boldsymbol{m} - \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \boldsymbol{u} \tag{6.18}$$

The controllability matrix of this timed net is the following:

$$\mathbb{C} = \begin{bmatrix} -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix}$$

Its rank being one, it has only one controllable pole (equal to -2) and one non controllable pole (equal to 0). A transformation matrix is:

$$\mathbb{Q} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

and, the corresponding $\mathbb{Q}$-canonical representation is:

$$\dot{\bar{\boldsymbol{m}}} = \begin{bmatrix} 0 & 0 \\ 1 & -2 \end{bmatrix} \bar{\boldsymbol{m}} - \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \boldsymbol{u}$$

6.6.3 Token conservation laws and controllable zero valued poles

In addition to those expressed by P-flows, other token conservation laws corresponding to zero valued poles can appear.

Example 6.30. *Let us consider the contPN system shown in Figure 6.8 with* $\boldsymbol{\lambda} = [\alpha,\beta,\gamma,\delta]^T$. *Clearly,* $\dot{m}_1+\dot{m}_2=\dot{m}_3+\dot{m}_4=\dot{m}_5+\dot{m}_6=0$ *are token conservation laws that correspond to zero valued uncontrollable poles (as mentioned in Example 6.24).*

If we fix m_2, m_3 *and* m_5 *as state variables then* m_1, m_4 *and* m_6 *are redundant. The linear dynamic system corresponding to the configuration* $\{(p_1,t_1),(p_2,t_2),(p_5,t_3),(p_6,t_4)\}$ *is:*

$$\begin{cases} \dot{m}_2 = -\beta\cdot m_2+\alpha\cdot(1-m_2) = -(\alpha+\beta)\cdot m_2+\alpha \\ \dot{m}_3 = -\beta\cdot m_2+\gamma\cdot m_5 \\ \dot{m}_5 = -\gamma\cdot m_5+\delta\cdot(1-m_5) = -(\gamma+\delta)\cdot m_5+\delta \end{cases}$$

Eliminating all variables in the right hand side:

$$-\frac{\beta}{\alpha+\beta}\cdot\dot{m}_2+\frac{\gamma}{\gamma+\delta}\cdot\dot{m}_5+\dot{m}_3 = \frac{\gamma\cdot\delta}{\gamma+\delta}-\frac{\alpha\cdot\beta}{\alpha+\beta} = q$$

Therefore, if $q=0$ *a new token conservation law appears introducing an additional zero valued pole (that it is not rooted in a P-flow):* $-\frac{\beta}{\alpha+\beta}\cdot m_2+\frac{\gamma}{\gamma+\delta}\cdot m_5+m_3 = constant$. *If* $q\neq 0$, *sooner or later the above configuration will be left. This is evident since at least one of the variables (*m_2, m_3 *or* m_5*) will grow (or decrease) while the system is in the configuration. This can also be deduced using the fact that the steady state flow has to be a T-semiflow of the net. Since it has only one minimal T-semiflow* $[1,1,1,1]^T$, *in steady state:* $f_1=f_2=f_3=f_4$.

$$f_1=f_2 \Longrightarrow \alpha\cdot m_1=\beta\cdot(1-m_1)\Longrightarrow f_1=\frac{\alpha\cdot\beta}{\alpha+\beta}$$

$$f_3=f_4 \Longrightarrow \gamma\cdot m_5=\delta\cdot(1-m_5)\Longrightarrow f_3=\frac{\gamma\cdot\delta}{\gamma+\delta}$$

$$f_1=f_3 \Longleftrightarrow \frac{\alpha\cdot\beta}{\alpha+\beta}=\frac{\gamma\cdot\delta}{\gamma+\delta}\Longleftrightarrow q=0$$

Thus, if $q\neq 0$ *the configuration will not be an equilibrium configuration.*

In this case, the system has the following poles: (0,0,0,-2,0,-2) (for $\boldsymbol{\lambda}=[1,1,1,1]^T$*) and three linearly independent P-flows. The fourth zero valued pole that appears in the configuration is given by a new token conservation law which depends on* $\boldsymbol{\lambda}$.

Obviously, the non controllable poles (P-flow related) appear in all the configurations. On the other hand, the controllable poles can have different values depending on the configuration. For example, if we consider the same system and the configuration $\{(p_1,t_1),(p_3,t_2),(p_5,t_3),(p_6,t_4)\}$, *the poles are: (0,0,0,-1,-1,-2).*

Example 6.31. *Let us see that for a specific value of* $\boldsymbol{\lambda}$, *additional token conservation laws and zero valued poles can appear. Let us consider the net in Figure 6.7 with* $\boldsymbol{\lambda}=[\alpha,\beta,\gamma]^T$ *and let us consider the configuration* $\{(p_4,t_1),(p_4,t_2),(p_3,t_3)\}$. *For place* p_2 *we can write:*

$$\dot{m}_2 = f_1-f_2 = \alpha\cdot\frac{m_4}{2}-\beta\cdot m_4 = m_4\cdot\left(\frac{\alpha}{2}-\beta\right)$$

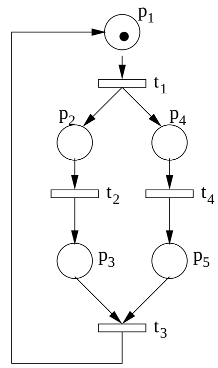


Figure 6.10: Choice-Free timed contPN system for Example 6.32.

1. $\frac{\alpha}{2} = \beta \Longrightarrow \dot{m}_2 = 0$. *In this situation, a new zero valued pole introduces a new token conservation law. For example, if* $\alpha = 2, \beta = 1, \gamma = 1$ *the poles of this configuration are:* $(0,0,0,-4)$.

2. $\frac{\alpha}{2} > \beta \Longrightarrow \dot{m}_2 > 0$. *Now, the marking of* p_2 *will increase and since the net is bounded, this configuration will be left sooner or later. Moreover, a positive pole appears. If* $\boldsymbol{\lambda} = [3,1,1]$ *the configuration poles are:* $(0,0,0.0981,-5.0981)$.

3. $\frac{\alpha}{2} < \beta \Longrightarrow \dot{m}_2 < 0$. *The marking of* p_2 *will decrease and the only solution to be an equilibrium configuration is to reach the deadlock (*m_4 *should be 0). All the controllable poles are negative. For* $\boldsymbol{\lambda} = [1,1,1]$ *they are* $(0,0,-0.1771,-2.8229)$.

So, other token conservation laws can appear depending on $\boldsymbol{\lambda}$ (here when $\frac{\alpha}{2} = \beta$) that introduce new zero valued but controllable poles.

6.6.4 Token conservation laws and controllable non zero valued poles

New token-invariant laws may appear depending on $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ (i.e. depending not only on the net structure as those derived from the P-semiflows), but also on $\boldsymbol{\lambda}$ and the precise marking $\boldsymbol{m}_0$. Let us present a simple case.

Example 6.32. *Consider now the contPN in Figure 6.10 with* $\boldsymbol{\lambda} = [\alpha, \beta, \delta, \gamma]^T$. *There exist two P-semiflows:* $m_1 + m_2 + m_3 = 1$ *and* $m_1 + m_4 + m_5 = 1$. *Then there are only three state variables, for example* m_1, m_3 *and* m_5. *The dynamic linear system associated with the configuration* $\{(p_1,t_1),(p_2,t_2),(p_3,t_3),(p_4,t_4)\}$ *is:*

$$\begin{cases} \dot{m}_1 = \delta \cdot m_3 - \alpha \cdot m_1 \\ \dot{m}_3 = -\delta \cdot m_3 + \beta \cdot (1 - m_1 - m_3) \\ \dot{m}_5 = -\delta \cdot m_3 + \gamma \cdot (1 - m_1 - m_5) \end{cases}$$

Nevertheless, if $\beta = \gamma$, $\dot{m}_3 - \dot{m}_5 = -\beta \cdot (m_3 - m_5)$. *Making a linear transformation in order to compute:* $\bar{m}_{35} = m_3 - m_5$, *then* $\dot{\bar{m}}_{35} = -\beta \cdot \bar{m}_{35}$. *If* $m_0[p_3] = m_0[p_5] \Longrightarrow \dot{\bar{m}}_{35} = 0$. *In this case, the pole is different from 0, and depends on* $\boldsymbol{m}_0$, *thus* $m_3 = m_5$ *is a token conservation law that is not rooted in a zero valued pole.*

6.7 Case study

Let us consider the manufacturing system sketched in Fig. 2.1 which consists in three machines ($M1$, $M2$ and $M3$) and two intermediate buffers ($Buffer_1$ and $Buffer_2$). Assume that each operation takes 1 time unit. Hence, the firing rate of all the transitions is 1.

This net has 5 P-semiflows ($\boldsymbol{y_1} = p_1 + p_2 + p_9 + p_{10} + M_1$, $\boldsymbol{y_2} = p_3 + p_{11} + Buffer_1$, $\boldsymbol{y_3} = p_4 + p_5 + p_{12} + p_{13} + M_2$, $\boldsymbol{y_4} = p_6 + p_{14} + Buffer_2$, $\boldsymbol{y_5} = p_7 + p_8 + p_{15} + p_{16} + M_3$) introducing in every configuration 5 uncontrolled zero valued poles. Computing the optimal steady-state (maximum flow) for the controlled contPN, the solution is: $\boldsymbol{w} = 0.2 \cdot \mathbf{1}$ and $\boldsymbol{u} = \mathbf{0}$.

This net has $\gamma = 256$ configurations and for each one Theorem 6.11 tells that all the equilibrium points have the same flow in steady state (i.e. 0.2) for the same control input $\boldsymbol{u} = \mathbf{0}$. Nevertheless, as expected, the equilibrium marking is not unique. For example, the configuration $\{(p_1, t_2), (p_2, t_3), (p_4, t_5), (p_5, t_6), (p_7, t_8), (p_8, t_9), (p_9, t_{11}), (p_{10}, t_{12}), (p_{12}, t_{14}), (p_{13}, t_{15}), (p_{15}, t_{17}), (p_{16}, t_{18}), (M_1, t_1), (M_1, t_{10}), (M_2, t_4), (M_2, t_{13}), (M_3, t_7), (M_3, t_{16})\}$ has:

$$rank \begin{bmatrix} \mathbf{\Pi} \\ \boldsymbol{B}_y^T \end{bmatrix} = 17 \tag{6.19}$$

Therefore, this configuration may contain an infinite number of equilibrium markings (Theorem 6.14). It is easy to see that the places corresponding to the P-semiflows given by the buffers ($\boldsymbol{y_2}$ and $\boldsymbol{y_4}$) can be loaded in any quantity greater than 0.2 and an equilibrium marking is obtained.

Computing the poles of this configuration we obtain 9 zero valued poles, three of them equal with $-2+i$, other three $-2-i$ and six equal with -1. Five of these zero valued poles are uncontrollable and are given by the P-semiflows and the other four are given by some token conservation laws given by the particular value of $\boldsymbol{\lambda}$ and the considered configuration. Anyhow, these are controllable and can be moved using an appropriate control law.

Chapter 7

Optimal control of continuous Petri nets

Summary

Our goal in this chapter is to find a control input optimizing a certain cost function that permits the evolution from an initial marking (state) to a desired steady-state. The solution we propose is based on a particular discrete-time representation of the controlled continuous Petri net system under infinite server semantics, as a certain linear constrained system. An upper bound on the sample period is given in order to preserve important information of the timed continuous net, in particular the positiveness of the markings. The reachability space of the sampled system in relation to autonomous continuous Petri nets is also studied.

7.1 Introduction

Steady state optimal control of contPN was studied in Section 6.5 and if all transitions can be controlled the problem can be solved in polynomial time. The solution is an optimal marking and an optimal control input in steady state.

In this chapter we assume that this steady state condition ($\boldsymbol{m_f}$, $\boldsymbol{w_f}$) is known and our problem is how to reach it (from a given $\boldsymbol{m}_0$) in a finite time while optimizing a given performance index.

Model Predictive Control (MPC) [47], also referred as *moving horizon control* or *receding horizon control*, is an advanced control method that has become an attractive control strategy. In the last years, many research groups have also worked on MPC of nonlinear systems. In the next sections we will show how these results can be applied in the case of contPN under infinite server semantics.

In the literature, the model predictive control is studied, in general, for discrete time systems. There is a few work regarding the continuous time systems, but finally, when this control is implemented on a computer the system should be discretized. For this reason, in this chapter we propose first a discrete time version of contPN systems with infinite server semantics based on a constrained linear representation.

Obviously, as happens in the classical linear systems, if the sampling period is too big, the discrete time approximation can be very bad. For contPN systems we give a bound of this sampling period, bound that will preserve the most important information: the non-negativity of the markings. For the discrete time contPN obtained with a good sampling period, we will study the reachability space and we will prove that under some conditions (that are more o less similar with the ones in previous chapter where the equivalence between the reachability of timed contPN and untimed contPN was studied) the reachability space of discrete time contPN is the same with the one of untimed contPN and with the one of timed contPN.

Using the discrete time approximation, we apply the MPC in both variants: implicit and explicit. Many simulations were performed and we show some results here. In the last part of the chapter, we study the asymptotic stability of a contPN with model predictive control scheme and we provide one that ensures this property.

7.2 Constrained linear representation of controlled timed continuous Petri nets

In order to apply the MPC scheme we rewrite the model used in the previous chapter in an equivalent form. The system in eq. (6.5) is a *piecewise linear* system with a dynamical constraint on the control input $\boldsymbol{u}$ that depends on the current value of the system state $\boldsymbol{m}$. For our control purposes, in this section we provide an alternative expression that takes the form of a simple *linear* system with dynamical constraints on the control input.

Proposition 7.1. *Any piecewise linear constrained model of the form* (6.5) *can be rewritten, by suitably defining a constant matrix* $\boldsymbol{G}$ *(see the proof of the proposition), as a linear constrained* *model of the form:*

$$\begin{cases} \dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{w}(\tau) \\ \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}(\tau) \\ \boldsymbol{m}(\tau) \end{bmatrix} \leq \boldsymbol{0} \\ \boldsymbol{w}(\tau) \geq \boldsymbol{0} \end{cases} \tag{7.1}$$

that we call continuous time controlled contPN *model, or* ct-contPN *model for short. The initial value of the state system is* $\boldsymbol{m}(0) = \boldsymbol{m}_0 \geq \boldsymbol{0}$.

Proof. The equivalence of the dynamic equations immediately follows by replacing $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$ in (7.1) being $\boldsymbol{f}(\tau)$ defined as in (2.14).

Concerning the constraints on the input, we first observe that, by virtue of (2.14), constraints in (6.5) can be rewritten as $\boldsymbol{0} \leq \boldsymbol{w}(\tau) \leq \boldsymbol{f}(\tau)$, i.e., $\forall j = 1, \cdots, n$, and at any marking $\boldsymbol{m}$,

$$0 \leq w_j \leq \lambda_j \min_{p_i \in {}^\bullet t_j} \left(\frac{m_i}{\boldsymbol{Pre}[p_i, t_j]} \right)$$

that is equivalent to the following set of equations

$$0 \leq w_j \leq \lambda_j \frac{m_i}{\boldsymbol{Pre}[p_i, t_j]} \qquad (\forall p_i \in {}^\bullet t_j).$$

All these equations can be combined as

$$\boldsymbol{0} \leq \boldsymbol{\Delta} \cdot \boldsymbol{w} \leq \boldsymbol{\Gamma} \cdot \boldsymbol{m}$$

where matrices $\boldsymbol{\Delta}$ $(q \times n)$ and $\boldsymbol{\Gamma}$ $(q \times m)$ have as many rows as there are "pre" arcs in the net, i.e., $q = \sum_{t \in T} |{}^\bullet t|$.

In particular, given a pre arc (p_i, t_j) the corresponding row of $\boldsymbol{\Delta}$ is the vector

$$\left[\underbrace{0 \quad \cdots \quad 0 \quad 1}_{j} \quad 0 \quad \cdots \quad 0 \right],$$

while corresponding row of $\boldsymbol{\Gamma}$ is the vector

$$\left[\underbrace{0 \quad \cdots \quad 0 \quad \frac{\lambda_j}{\boldsymbol{Pre}[p_i, t_j]}}_{i} \quad 0 \quad \cdots \quad 0 \right].$$

If we let

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{\Delta} & -\boldsymbol{\Gamma} \end{bmatrix}$$

we obtain the constraints in the last two equations of (7.1). □

The system in eq. (7.1) is a linear system with a *dynamic-matrix* equal to $\boldsymbol{0}$ and an *input matrix* equal to the *token flow matrix* of the contPN. Note however, that there is still a dynamical constraint on the system inputs that depends on the value of the system state $\boldsymbol{m}$.

7.3 On sampled (or discrete-time) continuous Petri nets models

Let us obtain a discrete-time representation of continuous-time contPN under infinite server semantics. Sampling should preserve the important information of the original model (for example the positiveness of the markings). This is directly studied in the next section through the equivalence of the reachability graph of the discrete-time model and the untimed model. In Section 6.4 the reachability space equivalence between continuous-time model and untimed model was studied and the equivalence was proved under the same conditions as in this case. Hence, the results in Section 6.4 together with those presented in this chapter provide as immediate conclusion that the reachability space of continuous-time and discrete-time are the same. In this section the discretization is defined together with a bound for the sampling period.

Definition 7.2. *Consider a ct-contPN as in eq.* (7.1) *and let* Θ *be a sampling period* ($\tau = k\cdot\Theta$). *The* discrete-time controlled contPN *or* dt-contPN $\langle \mathcal{N}, \lambda, \boldsymbol{m}_0, \theta\rangle$ *can be written as follows:*

$$\begin{cases} \boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta\cdot\boldsymbol{C}\cdot\boldsymbol{w}(k) \\ \boldsymbol{G}\cdot\begin{bmatrix} \boldsymbol{w}(k) \\ \boldsymbol{m}(k)\end{bmatrix} \leq \boldsymbol{0} \\ \boldsymbol{w}(k) \geq \boldsymbol{0} \end{cases} \tag{7.2}$$

The initial value of the state of this system is $\boldsymbol{m}(0) = \boldsymbol{m}_0 \geq \boldsymbol{0}$.

Example 7.3. *Let us consider the net system in Fig. 2.4 with* $\Theta = 1$. *Then the discrete-time representation is given by:*

$$\begin{cases} \boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \boldsymbol{C}\boldsymbol{w}(k) \\ w_1(k) - \frac{\lambda_1}{2}\cdot m_1(k) \leq 0 \\ w_1(k) - \frac{\lambda_1}{2}\cdot m_4(k) \leq 0 \\ w_2(k) - \lambda_2\cdot m_2(k) \leq 0 \\ w_2(k) - \lambda_2\cdot m_4(k) \leq 0 \\ w_3(k) - \lambda_3\cdot m_3(k) \leq 0 \\ \boldsymbol{w}(k), \boldsymbol{m}(k+1) \geq \boldsymbol{0} \end{cases} \tag{7.3}$$

thus

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & -\frac{\lambda_1}{2} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -\frac{\lambda_1}{2} \\ 0 & 1 & 0 & 0 & -\lambda_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -\lambda_2 \\ 0 & 0 & 1 & 0 & 0 & -\lambda_3 & 0 \end{bmatrix} \tag{7.4}$$

■

The reachability space of dt-contPN can be defined as follows.

Definition 7.4. *We denote* $RS^{dt}(\mathcal{N}, \boldsymbol{m}_0, \Theta)$ *the set of markings* $\boldsymbol{m} \in \mathbb{R}_{\geq 0}$ *such that there exists a finite input sequence* $\boldsymbol{w} = \boldsymbol{w}(0)\cdots\boldsymbol{w}(k)$ *and* $\boldsymbol{m}(0) \xrightarrow{\boldsymbol{w}(0)} \boldsymbol{m}(1) \xrightarrow{\boldsymbol{w}(1)} \boldsymbol{m}(2) \cdots \xrightarrow{\boldsymbol{w}(k-1)} \boldsymbol{m}(k) = \boldsymbol{m}$, *where* $\boldsymbol{0} \leq \boldsymbol{w}(j) \leq \boldsymbol{f}(j)$ $\forall j$, *and* $\boldsymbol{f}(j)$ *is the flow of the unforced system at time* $j\cdot\Theta$.

It is important to stress that, although the evolution of a sampled contPN occurs in discrete steps, *discrete time evolutions* and *untimed evolutions* are not necessarily the same. As an example, while an untimed net can be seen evolving sequentially, executing a single transition firing at each step, a dt-contPN may evolve in concurrent steps where more than one transition fires. We denote such a concurrent step as follows:

$$\boldsymbol{m}[\{t_{i_1}(\alpha_1), t_{i_2}(\alpha_2), \ldots, t_{i_k}(\alpha_k)\}\rangle \boldsymbol{m}'.$$

In unforced ct-contPN under infinite server semantics, the positiveness of the marking is ensured if the initial marking $\boldsymbol{m}_0$ is positive, because the flow of a transition goes to zero whenever one of the input places is empty [69].

In a dt-contPN, this is not always true.

Example 7.5. *Let us consider the net in Fig. 2.4, with* $\boldsymbol{m}_0 = [0.1, 5.9, 1, 5.9]^T$*,* $\boldsymbol{\lambda} = \mathbf{1}^T$*,* $\Theta = 2$*. Assume transitions* t_2 *and* t_3 *are stopped (*$w_2(0) = w_3(0) = 0$*), then* $m_1(1) = m_1(0) - 2\cdot\Theta\cdot w_1(0) = 0.1 - 4\cdot w_1(0)$*. But* $w_1(0)$ *is upper bounded by* $\frac{\lambda_1}{2}\cdot m_1(0) = 0.5\cdot 0.1 = 0.05$*. If the maximum value is chosen, then* $m_1(1)$ *will be negative!!!*

■

This can be avoided if the sampling period is small enough.

Proposition 7.6. *Let* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}, \Theta\rangle$ *be a dt-contPN system with* $\boldsymbol{m}_0 \geq \mathbf{0}$ *where the sampling period* Θ *is such that:*

$$\forall\, p \in P : \sum_{t_j \in p^\bullet} \lambda_j \Theta < 1. \tag{7.5}$$

Then the following statements hold.

1. *Any marking reachable from* $\boldsymbol{m}_0$ *is non negative, i.e.,*

$$RS^{dt}(\mathcal{N}, \boldsymbol{m}_0, \Theta) \subseteq \mathbb{R}^m_{\geq 0}.$$

2. *A place cannot be emptied with a finite sequence of firings, i.e., if* $\boldsymbol{m}[p] > 0$*, then* $\forall\, \boldsymbol{m}' \in RS^{dt}(\mathcal{N}, \boldsymbol{m}, \Theta)$ *it also holds* $\boldsymbol{m}'[p] > 0$*.*

Proof. Let us consider a place p_i with $p_i{}^\bullet = \{t_1, t_2, \cdots, t_j\}$ and $m_i(k) > 0$. Then:

$$\begin{aligned} m_i(k+1) &= m_i(k) + \Theta \boldsymbol{C}[i,\cdot]\,\boldsymbol{w}(k) \\ &\geq m_i(k) - \Theta(\lambda_1 + \lambda_2 + \cdots + \lambda_j) m_i(k) \\ &= m_i(k)\left(1 - \sum_{t_j \in p^\bullet} \lambda_j \Theta\right) > 0. \end{aligned}$$

□

In the rest of the chapter we will assume that all nets are sampled with a sampling period Θ that satisfies (7.5).

Proposition 7.7. *If* $\boldsymbol{m}$ *is reachable in a dt-contPN system* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}, \Theta\rangle$ *with* Θ *verifying* (7.5)*, then* $\boldsymbol{m}$ *is reachable in the underlying untimed contPN system* $\langle \mathcal{N}, \boldsymbol{m_0}\rangle$*, i.e.*

$$RS^{dt}(\mathcal{N}, \boldsymbol{m}_0, \Theta) \subseteq RS^{ut}(\mathcal{N}, \boldsymbol{m}_0).$$

Proof. In dt-contPN, transitions can fire concurrently and in order to prove that a marking is reached in the untimed contPN it is necessary to prove the existence of a sequence of transition firings leading to the same marking. This sequence exists due the fact that (7.5) implies $\boldsymbol{m}(k) - \boldsymbol{Pre} \cdot \Theta \cdot \boldsymbol{w}(k) \geq \boldsymbol{0}$ at any marking $\boldsymbol{m}(k)$. ☐

In general the converse of Proposition 7.7 is not true: in fact, the second item of Proposition 7.6 shows that in a dt-contPN with Θ satisfying (7.5) it is never possible to empty a place (only at the limit, thus timed contPN can be deadlocked only at the limit), while this may be possible in an untimed net system. As an example, in the untimed net system in Fig. 2.4 from the marking shown it is possible to fire $t_1(2)\,t_1(0.5)$, thus emptying place p_1. This marking is clearly not reachable on the same net system if we associate to it a firing rate vector and choose a sampling period Θ satisfying (7.5).

In the next section, two relaxations are studied: (1) considering in the untimed case only those sequences that never empty a marked place or (2) allowing the lim-reachable markings of the discrete-timed model. These relaxations are the same as in continuous-time case presented in the previous chapter. In fact we will prove that under any of these relaxations and with the sampling period as in (7.5), the reachability space of the discrete-time model will be the same as the reachability space of the continuous-time model.

7.4 Reachability "equivalence" between sampled and continuous models

Condition (7.5) can be seen like a kind of "Sampling Theorem": Θ should be small enough to maintain some properties as those in Proposition 7.6. But it does not mean that all signal information is preserved by sampling. The following result characterizes the reachability set of dt-contPN.

Lemma 7.8. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}, \Theta \rangle$ be a dt-contPN system. Assume that in the underlying untimed net system it is possible from $\boldsymbol{m}$ to fire the sequence $\boldsymbol{m}[t_j(\alpha)\rangle \boldsymbol{m}'$ and that for a certain $a > 1$, for all $p \in {}^{\bullet}t_j$ it holds $m'(p) \geq m(p)/a$.*

Then in the discrete time net system marking $\boldsymbol{m}'$ is reachable from marking $\boldsymbol{m}$ with a finite sequence of length

$$r = \left\lceil \frac{a}{\Theta \lambda_j} \right\rceil.$$

Proof. Let us first prove by induction that the firing of a sequence $[t_j(\alpha\Theta\lambda_j/a)\rangle$ can at least be repeated $r-1$ times in the discrete-time net.

(Basic step) It is immediate to observe that $t_j(\alpha\Theta\lambda_j/a)$ can be fired from $\boldsymbol{m}$, since $\Theta\lambda_j/a < 1$. The new marking is $\boldsymbol{m}_1 = (\alpha\Theta\lambda_j/a)\cdot\boldsymbol{m}' + (1-\alpha\Theta\lambda_j/a)\cdot\boldsymbol{m}$.

(Inductive step) Assume that at a given intermediate step $\boldsymbol{m_h} = \beta\boldsymbol{m}' + (1-\beta)\cdot\boldsymbol{m}$, with $0 < \beta < 1$. It can be observed that for all $p \in {}^{\bullet}t_j$, it holds $m_h(p) = \beta m'(p) + (1-\beta)m(p) \geq \beta\frac{m(p)}{a} + (1-\beta)\frac{m(p)}{a} = \frac{m(p)}{a}$, hence $t_j(\alpha\Theta\lambda_j/a)$ can be fired from $\boldsymbol{m_h}$, since $\Theta\lambda_j < 1$.

After $r-1$ firings $t_j(\alpha\Theta\lambda_j/a)$ can still be fired and it is sufficient to fire t_j for a quantity less or equal to that to reach $\boldsymbol{m}'$ in one step. ☐

According to the previous lemma, regardless of the initial token content in a place p, if an untimed sequence reduces the marking of p by at most a factor $1/a$, then an equivalent finite sequence exists in the dt-net system.

Theorem 7.9. *A marking $\boldsymbol{m}'$ is reachable in a dt-contPN $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}, \Theta \rangle$ system (with Θ satisfying* (7.5)*) iff it is reachable in the underlying untimed contPN system $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ with a sequence that never empties an already marked place.*

Proof. A sequence

$$\boldsymbol{m}[t_{i_1}(\alpha_1)\rangle \boldsymbol{m}_1[t_{i_2}(\alpha_2)\rangle \boldsymbol{m}_2 \cdots [t_{i_k}(\alpha_k)\}\rangle \boldsymbol{m}_k = \boldsymbol{m}'$$

never empties a marked place if the following condition is verified

$$(\forall j = 1, \ldots, k), (\forall p \in {}^\bullet t_{i_j})\ m_j(p) > 0 \tag{7.6}$$

(If) Applying the previous Lemma for each $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, ..., $\boldsymbol{m}_k$ implies that $\boldsymbol{m}'$ is reachable with a finite sequence.

(Only if) Assume there is a finite sequence that reaches $\boldsymbol{m}$ in the dt-contPN, then there exists an equivalent firing sequence for the untimed net system, according to Proposition 7.7. In the dt-contPN a place cannot be emptied with a finite sequence, according to Prop. 7.6 part 2. □

One may wonder what happens if a marking $\boldsymbol{m}$ is reachable in the untimed PN but there exists no sequence satisfying condition (7.6). In this case it can be proved that the marking is *lim*-reachable in the timed net, i.e., it is reachable with an unbounded sequence of steps. The result is formally stated in Theorem 7.10 by showing how such an infinite sequence may be determined.

Theorem 7.10. *If a marking $\boldsymbol{m}$ is reachable in the untimed contPN system $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$, then it is lim-rechable in a dt-contPN system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0}, \Theta \rangle$ with Θ satisfying* (7.5).

Proof. Assume that in the untimed net system

$$\boldsymbol{m}_0[t_{r_1}(\alpha_1)\rangle \boldsymbol{m}_1[t_{r_2}(\alpha_2)\rangle \boldsymbol{m}_2 \ldots [t_{r_k}(\alpha_k)\rangle \boldsymbol{m}_k = \boldsymbol{m},$$

and let us define $\sigma = t_{r_1}(\alpha_1)\, t_{r_2}(\alpha_2) \cdots t_{r_k}(\alpha_k)$.

We will prove that this sequence is equivalent to an infinite sequence $\sigma^1\sigma^2\cdots$ in which the marking of all input places of the fired transitions are reduced by each firing by at most a factor $1/2$. Thus, applying Lemma 7.8, it can be fired in the discrete time net. This infinite sequence will fire each transition in σ, but in a smaller amount, and repeat the process. It will be seen that the amount of firing of each transition converges to the value in σ.

For each round, the sequence is defined as

$$\begin{array}{rcl} \sigma^i & = & t_{r_1}(\beta_{i,1}\alpha_1)\, t_{r_2}(\beta_{i,2}\alpha_2) \cdots t_{r_k}(\beta_{i,k}\alpha_k) \\ \beta_{i,1} & = & 1/2^i,\ (i = 1, 2, \ldots) \\ \beta_{1,j} & = & 1/2^j, (j = 1, \ldots, k) \\ \beta_{i,j} & = & \frac{1}{2}\left(\sum_{l=1}^{i} \beta_{l,j-1} - \sum_{l=1}^{i-1} \beta_{l,j}\right), (i = 2, \ldots;\ j = 2, \ldots, k). \end{array}$$

It is proved in Proposition 6.6 that this infinite sequence converge to σ.

□

7.5 Optimal transient control via MPC

The basic idea of MPC is the following: at every time step, the control action is chosen solving an optimal control problem, minimizing a performance criterion over a future horizon. Only the first control command will be applied. After one time step other measurements will be got and the optimization problem is repeated. This is an on-line procedure and in many cases it is difficult (or even impossible) to implement because it requires the on-line solution of a linear or quadratic program (LP or QP, respectively), depending on the performance index.

MPC algorithms use different cost functions to obtain the control action. In this paper we consider the following standard quadratic form:

$$
\begin{aligned}
&J(\boldsymbol{m}(k), N) = \\
&\quad \big\{(\boldsymbol{m}(k+N) - \boldsymbol{m}_f)' \cdot \boldsymbol{Z} \cdot (\boldsymbol{m}(k+N) - \boldsymbol{m}_f) \\
&\quad + \sum_{j=0}^{N-1} \big[(\boldsymbol{m}(k+j) - \boldsymbol{m}_f)' \cdot \boldsymbol{Q} \cdot (\boldsymbol{m}(k+j) - \boldsymbol{m}_f) + \\
&\quad (\boldsymbol{w}(k+j) - \boldsymbol{w}_f)' \cdot \boldsymbol{R} \cdot (\boldsymbol{w}(k+j) - \boldsymbol{w}_f)\big]\big\}
\end{aligned}
\tag{7.7}
$$

where $\boldsymbol{Z}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite matrices.

The constraints are derived from the dt-contPN definition, and at every step the new marking should respect (7.2). Thus, at each step the following problem needs to be solved:

$$
\begin{array}{ll}
\min & J(\boldsymbol{m}(k), N) \\
\text{s.t.}: & \boldsymbol{m}(k+j+1) = \boldsymbol{m}(k+j) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k+j), \\
& \qquad\qquad\qquad\qquad j = 0, \ldots, N-1, \\
& \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}(k+j) \\ \boldsymbol{m}(k+j) \end{bmatrix} \leq \boldsymbol{0}, \qquad j = 0, \ldots, N-1, \\
& \boldsymbol{w}(k+j) \geq \boldsymbol{0}, \qquad\qquad j = 0, \ldots, N-1.
\end{array}
\tag{7.8}
$$

Now, using standard notation in the MPC framework, we say that (7.8) is a *finite time optimal control* (FTOC) problem if $N < \infty$; on the contrary we say that (7.8) is an *infinite time optimal control* (ITOC) problem if $N = \infty$. In such a case the first term of the performance index (7.7) should be neglected, or equivalently it should be assumed $\boldsymbol{Z} = \boldsymbol{0}$.

In [36] it has been proved that, if the region defined by the set of feasible state + input vectors is bounded and contains the final state + input in its interior, an ITOC problem may be reduced to a FTOC problem by appropriately choosing a finite value of N. In such a case the optimal control law after the finite horizon is taken equal to the unconstrained infinite horizon LQR problem with weights $\boldsymbol{Q}$ and $\boldsymbol{R}$. This guarantees both the constraint fulfillment for any time instant, and the asymptotic stability of the closed-loop system.

Note however that such a result is not applicable in many practical cases because controllers may be required to operate at the boundary of the feasible region. For instance, in the case of contPN the final marking may be null, that corresponds to minimize the inventory level. Then, if it often the case that we want to maximize the flow of certain transitions because this corresponds to maximize the production. As a result the final state and/or input are on the boundary and we cannot guarantee that an ITOC problem can be reduced to a FTOC problem.

We denote as *implicit* MPC the MPC control law computed solving on-line the optimization problem (7.8).

An alternative to *implicit* MPC has been proposed in [10], where the authors present a technique to compute *off-line* an *explicit* solution of the MPC control problem, based on multi-parametric linear programming (mp-LP) or quadratic programming (mp-QP). They split the maximum controllable set (i.e., all states that are controllable) into polytopes described by linear inequalities[1] in which the control command is described as a piecewise affine function of the state. Thus, the control law results in a *state feedback* control law.

In [10] it is shown how the state space partition and the affine control laws can be computed by means of multiparametric quadratic programming. The main results are presented in the following.

7.6 Explicit Model Predictive Control

In this section we recall the main features of eMPC. In particular, we show how a FTOC problem can be written as a multi-parametric programming problem and thus solved using appropriate efficient algorithms in this framework. This results in a state feedback control law based on a partition of the state space in polytopic regions.

Let us consider a discrete-time time-invariant system

$$x(k+1) = Ax(k) + Bu(k) \tag{7.9}$$

with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. Assume that the system (7.9) is subject to the constraint

$$Ex(k) + Lu(k) \leq M \tag{7.10}$$

for any $k \geq 0$.

Consider the following constrained FTOC problem

$$\begin{array}{rcll} J_N^*(x(0)) & = & \min\limits_{U_N} x_N' P x_N + \sum\limits_{k=0}^{N-1} u_k' R u_k + x_k' Q x_k & \\ s.t. & & Ex_k + Lu_k \leq M & k = 0,1,\dots,N-1 \\ & & x_N \in \mathcal{X}_{\text{set}} & \\ & & x_{k+1} = Ax_k + Bu_k & k = 0,1,\dots,N-1 \\ & & x_0 = x(0) & \end{array} \tag{7.11}$$

where N is the time horizon and $\mathcal{X}_{\text{set}} \subseteq \mathbb{R}^n$ is a terminal polyhedral region. In (7.11) we denote with

$$U_N \triangleq \left[u_0' \; u_1' \; \dots \; u_{N-1}'\right]' \in \mathbb{R}^s, \tag{7.12}$$

$s \triangleq mN$ the *optimization vector*; x_k is the state of the system at time k obtained starting from $x_0 = x(0)$ and applying the input sequence $u_0, \dots, u_{k-1}$; P, Q and R are the user-defined weighting matrices where $Q = Q' \succeq 0$, $R = R' \succ 0$, $P \succeq 0$. Finally, $x_N \in \mathcal{X}_{\text{set}}$ is a user-defined set-constraint on the final state which may be chosen such that the stability of the closed-loop system is guaranteed [55].

We define the N-step feasible set $\mathcal{X}_f^N \subseteq \mathbb{R}^n$ as the set of initial states $x(0)$ for which the FTOC problem (7.11) is feasible, i.e.,

$$\begin{array}{rl} \mathcal{X}_f^N = \{x(0) \in \mathbb{R}^n \mid & \exists (u_0,\dots,u_{N-1}) \in \mathbb{R}^s, \\ & Ex_k + Lu_k \leq M, \forall k = 1,\dots,N-1; \\ & x_N \in \mathcal{X}_{\text{set}}\}. \end{array} \tag{7.13}$$

[1] A bounded polyhedron $\mathscr{P} \subset \mathbb{R}^n$, $\mathscr{P} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{B}\}$ is called a *polytope*.

For a given initial state $x(0)$, problem (7.11) can be solved as a quadratic programming problem, but this type of on-line optimization may be prohibitive for control of fast processes.

By substituting

$$x_k = A^k x(0) + \sum_{j=0}^{k-1} A^k B u_{k-1-j},$$

problem (7.11) can be rewritten as

$$\begin{aligned} J_N^*(x(0)) \quad &= \quad \frac{1}{2}x'(0)Yx(0) + \min_{U_N} \frac{1}{2}U_N' H U_N + x'(0) F U_N \\ s.t. \quad & \quad\quad G U_N \leq W + E x(0) \end{aligned} \tag{7.14}$$

where $H = H' > 0$, H, F, Y, G, W are obtained from P, Q, R, (7.9), (7.10) (see [10]) for details).

Because problem (7.14) depends on $x(0)$, it can be solved as a multi-parametric program [10], i.e., considering $x(0)$ as a parameter, problem (7.14) can be solved for all parameters $x(0)$ to obtain a feedback solution, thus making this dependence *explicit*.

To this aim it is convenient to define

$$z \triangleq U_N + H^{-1} F' x(0), \tag{7.15}$$

$z \in \mathbb{R}^s$, and to transform (7.14) by completing squares to obtain the equivalent problem

$$\begin{aligned} J_z^*(x(0)) \quad &= \quad \frac{1}{2} z' H z \\ s.t. \quad & \quad\quad G z \leq W + S x(0) \end{aligned} \tag{7.16}$$

where $S \triangleq E + GH^{-1}F'$, $J_z^*(x(0)) = J_N^*(x(0)) - \frac{1}{2}x'(0)(Y - FH^{-1}F')x(0)$, and the parameter vector $x(0)$ appears only in the right hand side of the constraints.

Problem (7.16) is a multi-parametric programming problem that can be solved using appropriate recursive algorithms, presented in detail in [9, 10, 77]. Moreover, some of them are also implemented in the MATLAB toolbox MPT [44]. Now, considering $x(0)$ as a parameter, problem (7.16) can be solved for all parameters $x(0)$, thus obtaining a feedback solution with the following properties.

Theorem 7.11. [10, 14] Consider the constrained FTOC problem (7.11). Then, the set of feasible parameters $\mathscr{X}_f^N$ is convex, the optimizer $U_N^* : \mathscr{X}_f^N \to \mathbb{R}^s$ is continuous and piecewise affine, i.e.,

$$U_N^*(x(0)) = F_r x(0) + G_r \qquad \text{if} \qquad \begin{aligned} & x(0) \in \mathscr{P}_r = \{x \in \mathbb{R}^n \mid H_r x \leq K_r\}, \\ & r = 1, \ldots, N_R \end{aligned}$$

and the optimal cost $J_N^* : \mathscr{X}_f^N \to \mathbb{R}$ is continuous, convex and piecewise quadratic.

Thus, according to Theorem 7.11, the feasible state space $\mathscr{X}_f^N$ is partitioned into N_R polytopic regions, and a piecewise affine control law is univocally defined for each state $x \in \mathscr{X}_f^N$. When the system evolves, at each sampling instant k we need to determine the value of r such that $x(k) \in \mathscr{P}_r$, and apply the corresponding control law $F_r x(k) + G_r$.

Let us finally observe that the constrained ITOC problem, namely the optimal control problem in the case of $N = \infty$, can be solved using the above results provided that N is chosen large enough. The main theoretical results in this respect are due to Sznaier and Damborg with their pioneering work [74]. These results have been reconsidered and generalized later in [20, 65], and recently in [19].

The main advantage of the explicit approach is that the most burdensome part of the procedure is performed off-line, while the on-line part of the procedure simply consists in establishing in which region the current state is. However its applicability to real size cases is limited by two important facts. Firstly, the computational complexity of the off-line part highly increases with the length of the prediction horizon and with the order of the state space, becoming prohibitive for relatively modest values of these parameters. Moreover, the number of regions highly increases under the same circumstances, constituting a serious limitation to the on-line part of the procedure (because it makes it difficult to establish which control law should be applied).

Note that the explicit strategy can also be directly applied to the *piecewise* linear model. However the implementation of the control design for the constrained linear model is much simpler in particular because it does not require the knowledge of the state space *regions* where the different modes of the evolution occur, whose number highly increases with the number of places and transitions of the net.

7.7 Numerical examples

Using the control scheme presented in the previous sections, we consider two different numerical examples of discrete time contPN and make a detailed comparison among the results obtained with the above approaches.

The explicit solution has been computed using the Multi-Parametric Toolbox called MPT [44], a free and user-friendly MATLAB toolbox for design, analysis and deployment of optimal controllers for constrained linear and hybrid systems.

Implicit MPC has been implemented using GAMS [13] and MATLAB. In particular, the MINOS solver is utilized and the results of the optimization have also been compared with the results of other solvers.

7.7.1 First example

Let us consider the net system in Fig. 2.4 with $\boldsymbol{\lambda} = [1,1,1]^T$. Assume that the steady state (final) marking and control input are equal to $\boldsymbol{m}_f = [2.50, 3.25, 1.25, 2.50]^T$ and $\boldsymbol{w}_f = [1.25, 1.25, 1.25]^T$, respectively.

Note that even if the net has 4 places this is just a second order system. In fact the number of independent markings is equal to 2 because of the presence of two independent P-semiflows, namely $m_1 + m_2 + m_3 = 7$ and $m_1 + 4m_3 + m_4 = 10$.

Finally, we consider a quadratic performance index of the form (7.7) where $\boldsymbol{R} = \boldsymbol{I}$, $\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{Z} = 100 \cdot \boldsymbol{I}$.

Implicit MPC – FTOC problem

Table 7.1 summarizes the results obtained in the case of implicit MPC with initial marking equal to $\boldsymbol{m}_0 = [3,3,1,3]^T$, the sampling periods $\Theta = 0.1,\ 0.05,\ 0.01$, all satisfying the inequality (7.5), and different values of N.

$\Theta = 0.1$				$\Theta = 0.05$			
N	$\bar{J}$	comput. time [sec]	n_P	N	$\bar{J}$	comput. time [sec]	n_P
1	0.3409	0.0363	26	1	0.2359	0.0364	26
2	0.1794	0.0384	-	2	0.1795	0.0380	98
10	0.0851	0.0469	888	10	0.0822	0.0468	1436
20	0.0846	0.0614	2635	20	0.0810	0.0616	3202
$\Theta = 0.01$							
N	$\bar{J}$	comput. time [sec]	n_P				
1	0.0787	0.0372	12				
2	0.0787	0.0380	41				
10	0.0784	0.0449	651				
20	0.0782	0.0562	2489				

Table 7.1: The simulation results applied to contPN system in Fig. 2.4 with $\boldsymbol{m}_0 = [3,3,1,3]^T$.

The computational time is the average time in [sec] required to solve one QP problem in an Intel Pentium 4 at 3.20 GHz.

Finally, in order to compare the different evolutions, we compute the closed-loop infinite time horizon index multiplied by Θ, namely

$$\bar{J}(\boldsymbol{m}(0),\Theta) = \Theta \cdot \sum_{j=0}^{\infty} \Big[(\boldsymbol{m}(j) - \boldsymbol{m}_f)' \cdot \boldsymbol{Q} \cdot (\boldsymbol{m}(j) - \boldsymbol{m}_f) + (\boldsymbol{w}(j) - \boldsymbol{w}_f)' \cdot \boldsymbol{R} \cdot (\boldsymbol{w}(j) - \boldsymbol{w}_f) \Big] \tag{7.17}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the same weighting matrices used to compute the MPC controller.

From these, and other similar experiments, we can draw the following conclusions.

Firstly, the cost $\bar{J}$ is practically the same for $N = 10$ and $N = 20$, hence it is not necessary to increase very much the moving horizon to improve the solution. Note that for sufficiently large values of N this is not surprising. In fact, it is well known from classical systems' theory that there exists $\bar{N}$ such that for any initial state and any $N \geq \bar{N}$, the finite horizon controller is equal to the infinite horizon controller.

Secondly, we observe that, while for $\Theta = 0.1$ all the solutions are implementable in practice on this computer, this is not true in the other cases. In fact, the computational time to solve the QP problem becomes larger than the sampling period if N exceeds certain values. Some improvements can be done in order to reduce the computational times, e.g., rewriting the optimization problem as in [10], but these solutions have not been investigated here.

Explicit MPC – FTOC problem

The same numerical simulations have also been performed using the explicit approach. As already discussed above, in such a case we need to compute off-line an appropriate state space partition. As an example, in Fig. 7.1 we have reported the state space partition relative to the case of $\Theta = 0.1$ and $N = 10$.

In Table 7.1, columns 4, 8 and 12 summarize the number n_P of polytopic regions.

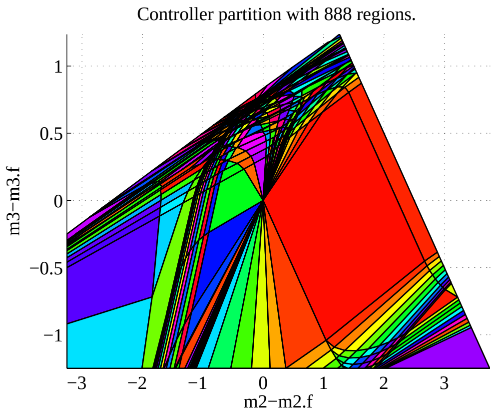


Figure 7.1: State space partition for the net system in Fig. 2.4 in the case of $N = 10$ and $\Theta = 0.1$.

Note that in the case of $\Theta = 0.1$ and $N = 2$ we have not been able to compute the explicit MPC controller because of numerical errors.

Obviously, when applicable, the explicit MPC provides the same control laws and thus the same values of the performance index $\bar{J}$ as the implicit MPC.

ITOC problem

The ITOC problem cannot be solved in this case. In fact the considered final input lays on the boundary of the region of feasible state + input vectors and it is not possible to determine a finite value of N in order to reduce the ITOC problem to a FTOC problem.

7.7.2 Second example

In order to show how the application of the explicit MPC approach may be unfeasible when the number of states increases, let us consider the net system in Fig. 7.2 with $\boldsymbol{\lambda} = [1, 1, 1, 1, 1, 1]^T$ and $\boldsymbol{m}_0 = [1, 1, 1, 0.5, 0, 0, 0.5, 0, 0]^T$. It is a 5th order system. In fact, even if the number of places is equal to 9, the number of independent markings is equal to 5 because of the presence of 4 independent P-semiflows: $m_1 + m_2 + m_5 + m_8 = 1$, $m_1 + m_3 + m_6 + m_9 = 1$, $m_4 + m_5 = 0.5$, $m_6 + m_7 = 0.5$.

Here, we assume as final marking and control input $\boldsymbol{m}_f = [0.25, 0.75, 0.7, 0.25, 0.25, 0.25, 0.25, 0.75, 0.8]^T$ and $\boldsymbol{w}_f = [0.25, 0.25, 0.25, 0.25, 025, 0.25]^T$, respectively.

We consider a quadratic performance index of the form (7.7) where $\boldsymbol{R} = \boldsymbol{I}$, $\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{Z} = 100 \cdot \boldsymbol{I}$. Finally, we assume $\Theta = 0.1$ that satisfies the inequality (7.5).

When applying the explicit MPC we found out that the number of polytopic regions is equal to $n_P = 743$ in the case of $N = 1$, while no result is obtained after 2 days of computations in the case of $N = 2$. Moreover, when we stopped computations the MPT toolbox had already computed almost 17000 regions.

As a result of this and other numerical examples we investigated, we conclude that the implicit MPC is often the only feasible solution even in the case of relatively low-order systems.

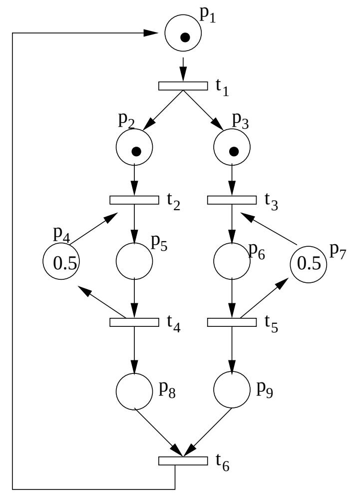


Figure 7.2: Timed continuous Marked Graph system.

7.8 Properties of the closed-loop system

In the above section we have shown how MPC (either implicit or explicit) may be used to control a contPN system in order to minimize a quadratic performance index that measures the distance from a desired state + input $(\boldsymbol{m_f}, \boldsymbol{w_f})$, e.g., a steady state.

In this section we want to investigate some properties of the resulting closed-loop system, such as feasibility and asymptotic stability.

7.8.1 Feasibility

In general, given an initial feasible state, i.e., the optimization problem is feasible at initial state (marking), there is no guarantee that the optimization problem we need to solve at each time step will remain feasible at all future time steps k, as the system might enter "blind alleys" where no solution to the optimization problem exists [10]. However, thanks to the particular structure of the constraints, in the case of contPN systems the following result can be proved, that guarantees the feasibility of (7.8) for any time step.

Proposition 7.12. *The optimization problem (7.8) is feasible for any* $\boldsymbol{m}(k) \geq \mathbf{0}$.

Proof. The solution $\boldsymbol{w}(k+j) = \mathbf{0}$ for $j = 0, 1, \dots, N-1$ is feasible. In fact,

$$\begin{aligned} \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}(k+j) \\ \boldsymbol{m}(k+j) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Delta} & -\boldsymbol{\Gamma} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{w}(k+j) \\ \boldsymbol{m}(k+j) \end{bmatrix} \\ &= -\boldsymbol{\Gamma} \cdot \boldsymbol{m}(k+j) \leq \mathbf{0} \end{aligned}$$

because (see Proposition 7.1) $\boldsymbol{\Gamma}$ is a matrix of non-negative numbers and $\boldsymbol{m}(k+j) = \boldsymbol{m}_k \geq \mathbf{0}$ for any $j = 1, \dots, N-1$. □

7.8.2 Asymptotic stability

The feasibility of (7.8) is obviously a desirable property but it does not ensure the convergence of the optimal solution to the desired state, that is our main requirement.

The following example clearly shows this.

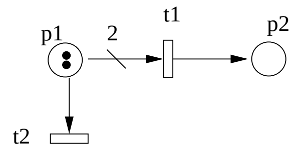


Figure 7.3: Example of an unstable contPN with basic MPC scheme.

Example 7.13. *Let us consider the net system in Fig. 7.3 with* $\boldsymbol{\lambda} = [1,5]^T$*. Let* $\Theta = 0.1$*,* $\boldsymbol{m}_f = [0,1]^T$ *and* $\boldsymbol{w}_f = [0,0]^T$*. Moreover, let* $\boldsymbol{Q} = \boldsymbol{Z} = \boldsymbol{R} = \boldsymbol{I}$ *and* $N = 1$*.*

The marking evolution of the system controlled with the MPC policy is presented in Fig. 7.4, that clearly shows that the desired marking is not reached. ■

In this section we investigate three different approaches in order to improve convergence. The first two approaches are quite standard in the literature, while, as far as we know, the latter approach has not been considered previously.

— **The first approach** consists in assuming that

$$\boldsymbol{w}(k+j) = \boldsymbol{0} \qquad \forall\ j = N, \cdots, \infty$$

and weighting the distance from the final marking not only for $j = 0,1,\cdots,N-1$ but for any $j = 0,1,\cdots,\infty$.

This is equivalent to consider an optimization problem of the form (7.8) where the matrix $\boldsymbol{Z}$ in the performance index is not an arbitrary positive definite matrix but $\boldsymbol{Z} = \boldsymbol{P}$, where $\boldsymbol{P}$ is the solution of the Lyapunov equation $\boldsymbol{P} = \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q}$, and $\boldsymbol{A} = \boldsymbol{I}$ in our case [60].

Unfortunately, this approach does not apply here because $\boldsymbol{Q}$ is a positive definite matrix so the equation $\boldsymbol{P} = \boldsymbol{P} + \boldsymbol{Q}$ is unfeasible.

— **The second approach** consists in assuming that

$$\boldsymbol{w}(k+j) = \boldsymbol{K}\boldsymbol{m}(k+j) \qquad \forall\ j = N, \cdots, \infty$$

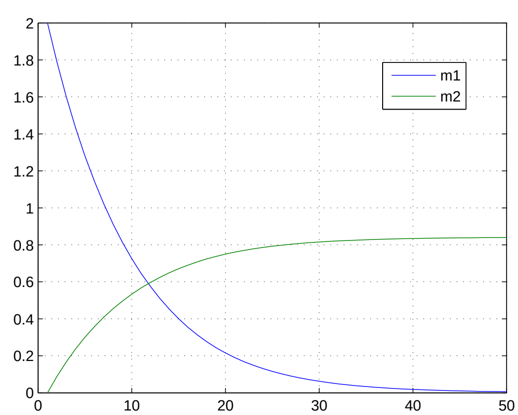


Figure 7.4: Marking evolution of the contPN in Fig. 7.3

and weighting the distance from the final marking not only for $j = 0, 1, \cdots, N-1$ but for any $j = 0, 1, \cdots, \infty$. In particular, matrix $\boldsymbol{K}$ is defined as in the unconstrained LQR problem with weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, namely

$$\begin{aligned} \boldsymbol{K} &= -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{K} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P} \boldsymbol{A}, \\ \boldsymbol{P} &= (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^T \boldsymbol{P} (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})) + \boldsymbol{K}^T \boldsymbol{R} \boldsymbol{K} + \boldsymbol{Q}, \end{aligned} \tag{7.18}$$

where in our case $\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{B} = \Theta \cdot \boldsymbol{C}$.

This is equivalent to consider an optimization problem of the form (7.8) where the matrix $\boldsymbol{Z}$ in the performance index is $\boldsymbol{Z} = \boldsymbol{P}$, where $\boldsymbol{P}$ is defined as in eq. (7.18).

In such a case, using results from the classical optimal control theory [65], we can guarantee convergence to the desired condition only if the region defined by the set of feasible state + input vectors is bounded and contains the final state + input in its interior.

As a consequence such an approach does not apply to most control problems within the framework of contPN, because the desired marking is often null and/or the desired flow is set to its maximum allowable value, thus we need to investigate for alternative procedures.

Note however that, if the final state + input is an interior point, and the moving horizon N is sufficiently large, this approach is surely the most convenient. In fact, it has the major advantage that the resulting strategy is indeed the optimal infinite horizon constrained LQR policy [10].

— **The third approach** we consider consists in forcing the marking at time $k+N$ to belong to the straight path $\boldsymbol{m}(k)$ — $\boldsymbol{m}_f$. In simple words, this is equivalent to add a terminal constraint of the form

$$\begin{cases} \boldsymbol{m}(k+N) = \alpha \cdot \boldsymbol{m}_f + (1-\alpha) \cdot \boldsymbol{m}(k) \\ 0 \le \alpha \le 1 \end{cases} \tag{7.19}$$

to the optimization problem (7.8), where α is a new decision variable.

Note that the addition of this constraint makes it necessary to solve a certain number of bilinear (rather than linear) programming problems when using explicit MPC [10]. In particular, bilinear problems have to be solved when computing the Chebychev centers of the polytopic regions, where both the initial state and α are unknown.

This approach revealed satisfactory in several numerical examples we considered even if we have been able to prove asymptotic stability only under certain assumptions, as detailed in the following.

Proposition 7.14. *Let us consider a contPN system. Let $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ be the initial and final markings, respectively, with $\boldsymbol{m}_0 > \boldsymbol{0}$ and $\boldsymbol{m}_f$ reachable from $\boldsymbol{m}_0$. Assume that the system is controlled using MPC with a terminal constraint of the form* (7.19) *and prediction horizon $N = 1$. Then the closed-loop system is asymptotically stable.*

Proof. We prove the statement in three steps. We first prove that if $\boldsymbol{m}_0 > \boldsymbol{0}$ then $\alpha > 0$ is feasible at any $k \ge 0$. Then, we define a quadratic function that we prove to be a Lyapunov function. Finally, we demonstrate that it is strictly decreasing.

- We first observe that by item (2) of Proposition 7.6, if $\boldsymbol{m}_0 > \boldsymbol{0}$ then $\boldsymbol{m}(k) > \boldsymbol{0}$ for any $k \ge 1$. Moreover, if $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}_0$ then it is also reachable from any marking in the straight path $\boldsymbol{m}_f$ — $\boldsymbol{m}_0$ being the full reachability space a convex region.

  Now, let us consider eq. (7.19) with $N = 1$. It holds $\boldsymbol{m}(k+1) = \alpha \cdot \boldsymbol{m}_f + (1-\alpha) \cdot \boldsymbol{m}(k)$. Being $\boldsymbol{m}_f$ reachable from $\boldsymbol{m}(k)$, then there exists $\boldsymbol{\sigma} \ge \boldsymbol{0}$ such that $\boldsymbol{m}_f = \boldsymbol{m}(k) + \boldsymbol{C} \cdot \boldsymbol{\sigma}$.

Thus, $\boldsymbol{m}(k+1) = \alpha \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{C} \cdot \boldsymbol{\sigma} + (1-\alpha) \cdot \boldsymbol{m}(k) = \boldsymbol{m}(k) + \boldsymbol{C} \cdot (\alpha\, \boldsymbol{\sigma})$. But there always exists $\alpha > 0$ such that $\alpha\, \boldsymbol{\sigma}$ can be fired at $\boldsymbol{m}(k)$ being $\boldsymbol{m}(k) > \mathbf{0}$.

- Now, without loss of generality we assume that $\boldsymbol{m_f} = \mathbf{0}$. In fact, if such is not the case, we can always redefine the state by translation, in order to meet such an assumption. Without loss of generality we may also assume that $\boldsymbol{C}$ in eq. (7.8) is full rank. This is always true if the net has no P-semiflows, otherwise we can always remove the independent P-semiflows in order make it true.

  Let
  $$V(\boldsymbol{m}(k)) = \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k)$$
  where $\boldsymbol{Z}$ is the weighting matrix in the performance index (7.7).

  Obviously, $V(\boldsymbol{m}(k)) \geq 0$ for any $\boldsymbol{m}(k) \neq \mathbf{0}$, being $\boldsymbol{Z}$ positive definite. Moreover, $V(\boldsymbol{m}(k+1)) \leq V(\boldsymbol{m}(k))$ for any $k \geq 0$. In fact, under the assumption that $\boldsymbol{m_f} = \mathbf{0}$, by constraint (7.19) it holds $\boldsymbol{m}(k+1) = (1-\alpha) \cdot \boldsymbol{m}(k)$. Thus,
  $$\begin{aligned} V(\boldsymbol{m}(k+1)) &= \boldsymbol{m}(k+1)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k+1) \\ &= (1-\alpha)^2 \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k) \\ &= (1-\alpha)^2 \cdot V(\boldsymbol{m}(k)) \leq V(\boldsymbol{m}(k)). \end{aligned}$$

- We now prove that $\forall\ k \geq 0$ the optimal solution of problem (7.8) leads to $\alpha > 0$.

  Let k be an arbitrary time instant.

  If $\alpha = 0$ then the performance index (7.7) is equal to
  $$J' = \boldsymbol{m}(k)^T \cdot \boldsymbol{Q} \cdot \boldsymbol{m}(k) + \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k).$$

  If $\alpha > 0$ (this is always possible by the first item of this proof), then the performance index (7.7) is equal to
  $$\begin{aligned} J'' &= \boldsymbol{m}(k)^T \cdot \boldsymbol{Q} \cdot \boldsymbol{m}(k) + \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) + \\ &\quad + \boldsymbol{m}(k+1)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k+1). \end{aligned}$$

  Being $\boldsymbol{m}(k+1) = (1-\alpha) \cdot \boldsymbol{m}(k)$, it holds
  $$J'' = \boldsymbol{m}(k)^T \cdot \boldsymbol{Q} \cdot \boldsymbol{m}(k) + \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) + \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k) - 2 \cdot \alpha \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k) + \alpha^2 \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k)$$
  and
  $$\begin{aligned} J'' - J' &= \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) + \alpha^2 \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k) - 2 \cdot \alpha \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k) \\ &= \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) + \alpha \cdot (\alpha - 2) \cdot \boldsymbol{m}(k)^T \cdot \boldsymbol{Z} \cdot \boldsymbol{m}(k). \end{aligned}$$

  But it is always possible to have $J'' < J'$ by appropriately choosing $\alpha > 0$, and this always occurs since we are minimizing the performance index. In fact, being
  $$\boldsymbol{m}(k+1) = (1-\alpha) \cdot \boldsymbol{m}(k) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k),$$
  then
  $$\boldsymbol{m}(k) = -\frac{\Theta}{\alpha} \cdot \boldsymbol{C} \boldsymbol{w}(k).$$

Therefore

$$
\begin{aligned}
J'' - J' &= \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) - \frac{\Theta^2 \cdot (2-\alpha)}{\alpha} \cdot \boldsymbol{w}(k)^T \cdot \boldsymbol{C}^T \cdot \boldsymbol{Z} \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k) \\
&= \boldsymbol{w}(k)^T \cdot \boldsymbol{R} \cdot \boldsymbol{w}(k) - \boldsymbol{w}(k)^T \cdot \left[\left(\frac{2}{\alpha} - 1 \right) \cdot \Theta^2 \cdot \boldsymbol{C}^T \cdot \boldsymbol{Z} \cdot \boldsymbol{C} \right] \cdot \boldsymbol{w}(k) < 0
\end{aligned}
$$

if α is small enough and $\boldsymbol{C}^T \cdot \boldsymbol{Z} \cdot \boldsymbol{C}$ is positive definite. But $\boldsymbol{C}^T \cdot \boldsymbol{Z} \cdot \boldsymbol{C}$ is always positive definite because $\boldsymbol{Z}$ is positive definite by definition and $\boldsymbol{C}$ is a full rank matrix by assumption.

□

Remark 7.15. *In general $\boldsymbol{m}(0) > \boldsymbol{0}$ is not a strict requirement in the above proposition. It is sufficient to assume that for any $k \geq 0$ the optimization problem (plus terminal constraint) admits $\alpha > 0$ as a solution. Physically this means that we can move along the straight line $\boldsymbol{m}(0)$ — $\boldsymbol{m_f}$. However, since in general it is difficult to verify such a condition, for simplicity of presentation we prefer to claim the statement of Proposition 7.14 providing a condition on $\boldsymbol{m}(0)$.*

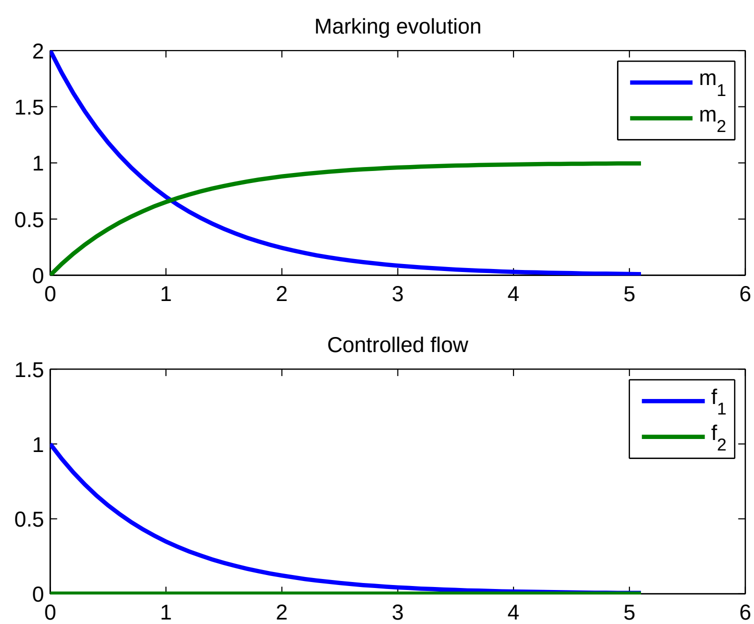


Figure 7.5: Marking and controlled flow evolution of the contPN system with terminal contrsints.

Example 7.16. *Let us go back to the Ex. 7.13 considering now the MPC scheme with terminal constraints of the form* (7.19)*. Observe that the initial marking has a null component but it is possible to follow the direction from the beginning and according to the previous remark and Prop. 7.14 the desired marking $\boldsymbol{m}_f = [0,1]^T$ is reached. The marking evolution is sketched in Fig. 7.5 and can be seen that $\boldsymbol{m}_f$ is reached.*

7.9 Conclusions

In this chapter we have considered *timed contPN* under infinite server semantics and we have studied the transitory control. Our problem is to reach a final marking optimizing a performance index. First we give a linear constrained form (eq. (7.1)) that help us to obtain the discrete time representation of contPN systems with infinite server semantics. For this representation we have given a bound on the sampling period, bound that preserves the reachability conditions (in particular non-negativity of markings).

Some relations between the reachability spaces of the sampled system and the untimed one are given and it is shown under which conditions these two spaces are equivalent. Using the discrete-time linear model, an optimal control is applied based on MPC. Both variants, implicit and explicit, are studied and compared with many simulations, here we have shown only few of them. And, in the last part, the feasibility and convergence of this control scheme is studied providing a particular control law that ensures the asymptotic stability.

Chapter 8

Concluding remarks

Continuous Petri nets can be viewed -but not only- as an approximation of discrete Petri nets, introduced to deal with the state explosion problem frequently appearing in discrete event systems. They relax the firing of a transition to a real non negative amount and it is not limited as in discrete systems to a natural quantity. Therefore, the marking of a continuous Petri net system is not a vector of integers but of non negative real values. Unfortunately, not all discrete systems can be fluidified and the properties of the original discrete system are not always preserved by its continuous approximation.

This thesis mainly deals with timed contPN systems. The untimed contPN systems are considered only for the state-estimation problem where the results from discrete Petri nets are adapted for the continuous systems. For the timed systems the flow through a fluidified transition can be defined in many ways. The most used in literature are constant and variable (enabling proportional) speed, which can be seen as some kind of finite and infinite server interpretations of the transitions behavior, and derived from stochastic (discrete) Petri nets.

For discrete PNs infinite server semantics is *more general*, since it can simulate finite server semantics. However, the continuous approximation of this simulation model under infinite server semantics is not equivalent to using finite server semantics in the original fluidified model. In the continuous case the two semantics are in fact related to different relaxations of the model. Since the two semantics are different, the immediate question is, given a net, which continuous semantics is better? Up to now, the first reference to this problem we were able to find in the literature is a remark in [2], where the authors mention having observed that in several cases *infinite server semantics* provides a very accurate approximation of discrete PNs.

One of the goals of chapter 3 is to provide some results about the use of these semantics. It can be seen that it is not possible to say that one of them is always better than the other. Hence, we have considered a class of nets, hopefully with a significant modeling power from a practical point of view. The class is called mono-T-semiflow reducible. Focusing on live and bounded systems, this class includes equal conflict nets, which is a superset of the classes of free-choice, choice-free, weighted T-systems and marked graphs nets. It is proved that for mono-T-semiflow reducible nets infinite server semantics provides a better approximation of the steady state throughput of the system, if all possible steady-states contain the support of a P-semiflow.

Another good indicator of the practical interest of a class of net models is the kind of properties it verifies. In chapter 3 the monotonicity in steady-state with respect to the firing rate of the transitions and with respect to the initial marking is studed. This property is checked for the continuous timed systems under infinite server semantics. Monotonicity is a very desired property in the production systems since replacing a slower machine with a faster one or increasing the number of the resources, the production should increase to justify the investments. We have proved that if all possible equilibrium configurations contain the support of a P-semiflow then the net system exhibits these monotonicity properties.

In discrete models, an approximate way to deal with stiffness was to drastically classify the transitions into *immediate* and *timed* (thus markings into *vanishing* and *tangible*). The same kind of classification can be done for contPN. However, removing immediate transitions at net level becomes a crucial issue in contPNs, because the semantics of timed continuous net models is directly expressible as ODEs with minimum operators, for which reasonable solutions are frequently obtained by means of numerical integration. The basic goals of the last part of chapter 3 are: (1) reduce the number of minimum operators both for any timing (implicitness in the autonomous model) and for a particular timing (implicitness in the timed model); and (2) provide the semantics for immediate transitions and some rules to reduce their number, since they are not easy to deal with in simulation or numerical computations.

State estimation is a fundamental issue in system theory. Reconstructing the state of a system from available measurements may be considered as a self-standing problem, or it can be seen as a pre-requisite for solving a problem of a different nature, such as stabilization, state-feedback control, diagnosis, filtering, and others. Despite the fact that the notions of state estimation, observability and observer are well understood in time driven systems, in the area of discrete event and of hybrid systems there are relatively few works addressing these topics and several problems are still open.

In chapter 4 timed contPN systems under infinite server semantics and some observability problems are considered. First, some characterizations of the observability for the general nets are given based on the literature on hybrid systems. Here, observability in infinitesimal time is studied and necessary and sufficient conditions for contPN systems are given. If a net system is structurally observable then it is observable for any values of firing rates associated to the transitions. Moreover if it is generic observable then it is observable for "almost" any values of firing rates. In the chapter, for some subclasses, the set of places that makes a timed net system structurally or generically observable is determined.

If some places have associated a measurement cost, an interesting problem is to determine the set of places with minimum cost that make the timed net system observable. This problem is studied in the last part of chapter 4 where a tree based algorithm is provided. This algorithm reduces the practical complexity of the problem, that is NP-complete using some information of the net. In particular, the net is split into subnets from which only one place can belong to the optimal solution.

Starting by showing that the results in chapter 4 are impossible to apply in case of timed contPN systems with finite server semantics, in chapter 5, state estimation of this class is studied. As well, this problem is studied for discrete PNs. Assuming that the firing amount in which some transitions fire can be observed the problem is to determine the set of consistent markings (markings in which the system can be) with this observation. Firstly, it is shown how the results obtained for discrete Petri nets naturally extended in the untimed contPN case. Then, for timed contPN this problem is started and here it is assumed that no observation is available, hence the problem is to determine the reachability space at some time.

Assuming that certain discrete event systems usually work close to congestion, and having evidence of the gains that in certain cases are obtained by fluidification, chapter 6 deals with some control problems of timed continuous PNs under infinite server semantics for which the continuous model is a multilinear switched dynamic system.

The control problem is simply based on the idea of *slowing down* the firing flow of transitions. It is shown that if all transitions are controllable then the reachability space of the timed net system coincides with the reachability space of the untimed one, maybe considered at the limit. The determination of the steady state is impossible in many cases because it is an undecidible problem [37]. Anyhow, some characterizations of possible steady-states are given in chapter 6. For some classes it is proved that even if they are more than one, they have the same flow in steady-state. The second main contribution of chapter 6 is the computation of an optimal steady state control reference that in the case in which all transitions are controllable, this problem is solved in polynomial time, using a LPP. Otherwise, a branch and bound algorithm can be used, similar with the one to compute the performance bounds.

In many cases, the regulation of the system is done to a point that is not on the boundary, hence the dynamical constraints on the input are not active. In this case a better understanding of the behavior of contPN using the classical results of linear system can be useful. In the last part of chapter 6, the poles location of the linear systems that can govern the evolution of a contPN is studied. It is shown that net systems generate different token conservation laws, some of them leading to uncontrollability. Some conservation laws are generated by the P-flows (which depend only on the net structure) and zero valued poles appear in the *uncontrollable* part of the system. Other zero valued *controllable* poles are related to conservation laws that depend on the net structure, the firing rates and the token load of P-(semi)flows. Finally, some *controllable* non zero poles may generate token conservation laws for particular values of $\boldsymbol{m}_0$.

In chapter 7, we have considered *timed contPN* under infinite server semantics and rewritten the non-linear system in a linear constrained form. The linear constrained system is then discretized and we provide a *Sampling theorem* giving an upper bound on sampling period. The purpose of the Sampling theorem presented here is to preserve reachability conditions (in particular non-negativity of markings), not to reconstruct the original signal from the sampled one. In practice, the sampling rate may need to be higher (like in Nyquist-Shannon sampling theorem) if signal reconstruction is required. But this is an open topic to be considered in future.

The reachability space of the sampled system is studied later and some relations between this space and the space of the underlying untimed contPN are provided. Then, on the basis of the constrained discrete-time positive linear model of the system, we derived optimal control laws based on MPC. In particular, we investigated the possibility of using both implicit and explicit control. Some aspects regarding the convergence of MPC are studied, and for a particular control law asymptotic stability is guaranteed.

An interesting issue is to extend the results presented here to more general contPN classes. For example, the performance monotonicity that has been studied only for mono T-semiflow reducible nets, optimal observability is studied only for Join Free, Continuous Equal Conflict and Attribution Free nets; some results of the state estimation of systems with finite server semantics are restricted to the conservative and consistent nets; an asymptotic stable MPC scheme is given only if the timing horizon is one. Observability of systems with infinite server semantics is studied only in infinitesimal time and aspects of finite time observability and infinite time represent also some interesting problems. State estimation of systems with finite server semantics has been introduced but the problem deserve more attention in the future. Others (sub)optimal control scheme can be purposed guaranteing the reachability of the desired marking.

Bibliography

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995. [cited at p. 8, 24, 25, 50]

[2] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8(1):159–188, 1998. [cited at p. 1, 17, 82, 87, 139]

[3] M. Babaali and M. Egerstedt. On the observability of piecewise linear systems. In *43rd IEEE Conference on Decision and Control (CDC 2004)*, pages 26–31, Paradise Island, Bahamas, December 2004. [cited at p. 55]

[4] M. Babaali and G. J. Pappas. Observability of switched linear systems in continuous time. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005,*, volume 3414 of *Lecture Notes in Computer Science*, pages 103–117, Zurich, Switzerland, 2005. Springer. [cited at p. 55]

[5] F. Balduzzi, A. Di Febbraro, A. Giua, and C. Seatzu. Decidability results in first-order hybrid Petri nets. *Journal of Discrete Event Dynamic Systems*, 11(1–2):41–58, 2001. Special Issue on Hybrid Petri nets. [cited at p. 1]

[6] F. Balduzzi, A. Giua, and C. Seatzu. Modelling and simulation of manufacturing systems with first-order hybrid Petri nets. *Int. J. of Production Research*, 39(2):255–282, 2001. Special Issue on Modelling, Specification and analysis of Manufacturing Systems. [cited at p. 1, 87]

[7] F. Balduzzi, G. Menga, and A. Giua. First-order hybrid Petri nets: a model for optimization and control. *IEEE Trans. on Robotics and Automation*, 16(4):382–399, 2000. [cited at p. 1, 17, 18, 82, 87, 88]

[8] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. L. Sangiovanni-Vincentelli. Observability for hybrid systems. In *Proc. 42nd IEEE Conference on Decision and Control*, Hawaii, USA, December 2003. [cited at p. 55]

[9] M. Baotić. An efficient algorithm for multi-parametric quadratic programming. Technical Report AUT02-04, Automatic Control Laboratory, ETH Zurich, 2002. [cited at p. 128]

[10] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002. [cited at p. 4, 127, 128, 130, 132, 134]

[11] A. Bemporad, F.D. Torrisi, and M. Morari. Performance analysis of piecewise linear systems and model predictive control systems. In *Proc. of the 39th IEEE Conference on Decision and Control*, pages 4957–4962, Sydney, Australia, 2000. [cited at p. 4]

[12] G. Berthelot. Checking properties of nets using transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 1986. [cited at p. 44]

[13] R.F. Boisvert, S.E. Howe, and D.K. Kahaner. Gams: A framework for the management of scientific software. *ACM Transactions on Mathematical Software*, 11(4):313–355, 1985. [cited at p. 129]

[14] F. Borrelli. Constrained optimal control of linear and hybrid systems. *Lectures Notes in Control and Information Sciences*, 290, 2003. [cited at p. 128]

[15] M.P. Cabasino, A. Giua, C. Mahulea, and C. Seatzu. State estimation of untimed and timed continuous petri nets. Research report, University of Cagliari, Italy, 2007. Submitted for publication. [cited at p. 4]

[16] J. Campos, G. Chiola, and M. Silva. Ergodicity and throughput bounds of Petri net with unique consistent firing count vector. *IEEE Trans. on Software Engineering*, 17(2):117–125, 1991. [cited at p. 112]

[17] J. Campos and M. Silva. Structural techniques and performance bounds of stochastic Petri net models. In G. Rozenberg, editor, *Advances in Petri Nets 1992*, volume 609 of *Lecture Notes in Computer Science*, pages 352–391. Springer, 1992. [cited at p. 28, 29]

[18] C. G. Cassandras. *Discrete Event Systems. Modeling and Performace Analysis.* Asken Associates, 1993. [cited at p. 28, 29]

[19] L. Chisci and G. Zappa. Fast algorithm for a constrained infinite horizon LQ problem. *Int. J. of Control*, 72(11):1020–1026, 1999. [cited at p. 129]

[20] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems and Control Letters*, 29(3):121–130, 1996. [cited at p. 129]

[21] G. Cohen, S. Gaubert, and J. P. Quadrat. Algebraic system analysis of timed petri nets. In J. Gunawardena, editor, *Idempotency: Collection of the Isaac Newton Institute.* Cambridge University Press, 1998. [cited at p. 2]

[22] P. Collins and J.H. van Schuppen. Observability of piecewise-affine hybrid systems. In R. Alur and G.J. Pappas, editors, *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC 2004,*, volume 2993 of *Lecture Notes in Computer Science*, pages 265–279, Philadelphia, USA, 2004. Springer. [cited at p. 55, 56, 57]

[23] C. Commault, J.M. Dion, and D.H. Trinh. Observability recovering by additional sensor implementation in linear structured systems. In *Proceedings of the* 44^{th} *IEEE Conference on Decission and Control 2005*, pages 7193–7197, Seville, Spain, December 2005. [cited at p. 69, 70]

[24] Thomas H. Cormen, E. Leiserson, Charles, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Press, 1990. [cited at p. 64]

[25] D. Corona, A. Giua, and C. Seatzu. Marking estimation of Petri nets with silent transitions. *IEEE Transactions on Automatic Control*, 2007. Accepted for publication. [cited at p. 82, 84, 86, 87]

[26] J.G. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proc. of the IEEE INFOCOM*, pages 2:556–564, 2000. [cited at p. 1]

[27] R. David and H. Alla. Continuous Petri nets. In *Proc. 8th European Workshop on Application and Theory of Petri Nets*, Zaragoza, Spain, 1987. [cited at p. 1]

[28] R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets.* Springer-Verlag, 2005. [cited at p. 1, 2, 3, 8, 11, 17, 23]

[29] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat. *Practice of Petri Nets in Manufacturing.* Chapman & Hall, 1993. [cited at p. 8]

[30] 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. [cited at p. 69]

[31] M.R. Garey and D.S. Johnson. *Computers and Interactability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979. [cited at p. 71]

[32] B. Gaujal and A. Giua. Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica*, 40(9):1505–1516, 2004. [cited at p. 2]

[33] A. Giua, C. Mahulea, L. Recalde, C. Seatzu, and M. Silva. Optimal control of timed continuous Petri nets via explicit MPC. In *Proc. of* 2^{nd} *Multidisciplinary International Symposium on Positive Systems: Theory and Applications*, Lecture Notes in Control and Information Sciences, pages 383–390. Springer Berlin / Heidelberg, 2006. [cited at p. 4]

[34] A. Giua, C. Mahulea, L. Recalde, C. Seatzu, and M. Silva. Optimal Control of timed continuous Petri Nets via model predictive control. In *WODES'06:* 8^{th} *International Workshop on Discrete Event Systems*, pages 235–241, Ann Arbor, USA, July 2006. [cited at p. 4]

[35] A. Giua and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. In *Proc. IEEE* 44^{rd} *Int. Conf. on Decision and Control*, December 2005. A corrected version can be found on: `http://www.diee.unica.it/~giua/PAPERS/CONF/05cdc-ecc_a_up.pdf`. [cited at p. 86]

[36] P.O. Gutman and M. Cwikel. Admissible sets and feedback control for discrete-time linear dynamical systems with bounded control and states. *IEEE Transactions on Automatic Control*, 31(4):373–376, 1986. [cited at p. 126]

[37] S. Haddad, L. Recalde, and M. Silva. On the computational power of timed differentiable Petri nets. In E. Asarin and P. Bouyer, editors, *Formal Modeling and Analysis of Timed Systems, 4th Int. Conf. FORMATS 2006*, volume 4202 of *LNCS*, pages 230–244, Paris, 2006. Springer. [cited at p. 1, 20, 141]

[38] S. Haddad, L. Recalde, and M. Silva. On the computational power of timed differentiable Petri nets. Technical Report RR-06-05, Universidad de Zaragoza, Zaragoza, Spain, 2006. [cited at p. 32]

[39] J. Júlvez. *Algebraic Techniques for the Analysis and Control of Continuous Petri nets.* PhD thesis, Universidad de Zaragoza, 2004. [cited at p. 2]

[40] J. Júlvez, E. Jiménez, L. Recalde, and M. Silva. On observability in timed continuous Petri net systems. In 1^{st} *Conf. on Quantitative Evaluation of Systems (QEST) 2004*, Twente, The Netherlands, September 2004. [cited at p. 3, 53, 63, 67]

[41] 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} *International Conference on Application and Theory of Petri Nets (ICATPN 2003)*, volume 2679 of *Lecture Notes in Computer Science*, pages 221–240. Springer, Eindhoven, The Netherlands, June 2003. [cited at p. 13, 14, 16]

[42] J. Júlvez, L. Recalde, and M. Silva. Steady-state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605–616, 2005. [cited at p. 1, 3, 26, 28, 29, 30, 33, 37, 49, 67, 107, 108, 109, 110, 111]

[43] J. Júlvez, L. Recalde, and M. Silva. Deadlock-freeness analysis of continuous mono-T-semiflow Petri nets. *IEEE Trans. on Automatic Control*, 51(9):1472–1481, 2006. [cited at p. 16, 32]

[44] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. [cited at p. 128, 129]

[45] D. G. Luenberger. *Introduction to Dynamic Systems. Theory, Models, and Applications.* John Wiley & Sons, 1979. [cited at p. 98]

[46] D.G. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602, December 1971. [cited at p. 54, 55]

[47] J. M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002. [cited at p. 120]

[48] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. On sampling continuous timed PNs: reachability "equivalence" under infinite servers semantics. In 2^{nd} *IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 37–43, Alghero, Italy, June 2006. [cited at p. 4]

[49] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous petri nets. Research report, Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna, 1, 50018 Zaragoza, Spain, 2006. Submitted for publication. [cited at p. 4]

[50] C. Mahulea, A. Ramírez, L. Recalde, and M. Silva. Steady state control, zero valued poles and token conservation laws in continuous net systems. In *Workshop on Control of Hybrid and Discrete Event Systems*, Miami, USA, June 2005. J.M. Colom, S. Sreenivas and T. Ushio, eds. [cited at p. 4]

[51] C. Mahulea, A. Ramírez, L. Recalde, and M. Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Transactions on Automation Science and Engineering*, 2007. to appear. [cited at p. 4]

[52] C. Mahulea, L. Recalde, and M. Silva. Optimal observability for continuous Petri nets. In 16^{th} *IFAC World Congress*, Prague, Czech Republic, July 2005. [cited at p. 4, 65]

[53] C. Mahulea, L. Recalde, and M. Silva. On performance monotonicity and basic servers semantics of continuous Petri nets. In *WODES'06:* 8^{th} *Workshop on Discrete Event Systems*, pages 345–351, Ann Arbor, USA, July 2006. [cited at p. 4]

[54] C. Mahulea, L. Recalde, and M. Silva. On performance monotonicity and basic servers semantics of continuous Petri nets. 2007. submitted for publication. [cited at p. 4]

[55] D.Q. Mayne and S. Racović. Model predictive control of constrained piecewise affine discrete-time systems. *Int. J. of Robust and Nonlinear Control*, 13(3):261–279, 2003. [cited at p. 127]

[56] J. C. Mugarza, H. Camus, J.-C. Gentina, E. Teruel, and M. Silva. Reducing the computational complexity of scheduling problems in petri nets by means of transformation rules. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'98)*, pages 19–25, October 1998. [cited at p. 44]

[57] T. Murata. State equation, controllability, and maximal matchings of Petri nets. *IEEE Trans. on Automatic Control*, 22(3):412–416, 1977. [cited at p. 113]

[58] K. Ogata. *Discrete-Time Control Systems, 2nd. ed.* Prentice Hall, 1995. [cited at p. 54, 55]

[59] C.A. Petri. *Kommunikation mit Automaten (Communication with Automata)*. PhD thesis, Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962. Second Edition:, New York: Griffiss Air Force Base, Technical Report RADC-TR-65–377, Vol.1, 1966, Pages: Suppl. 1, English translation. [cited at p. 1]

[60] J.B. Rawlings and K.R. Muske. The stability of constrained receding-horizon control. *IEEE Transaction on Automatic Control*, 38:1512 – 1516, 1993. [cited at p. 133]

[61] L. Recalde. *Structural Methods for the Design and Analysis of Concurrent Systems Modeled with Place/Transition Nets*. PhD thesis, Universidad de Zaragoza, 1998. [cited at p. 2]

[62] L. Recalde, C. Mahulea, and M. Silva. Improving analysis and simulation of continuous Petri nets. In 2^{nd} *IEEE Conference on Automation Science and Engineering*, pages 7–12, Shanghai, China, October 2006. [cited at p. 1, 4]

[63] 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. [cited at p. 28]

[64] 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. [cited at p. 12, 14, 16, 33, 108]

[65] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Transaction on Automatic Control*, 43(8):1163 – 1169, 1998. [cited at p. 129, 134]

[66] M. Silva. *Las Redes de Petri: en la Automática y la Informática*. AC, 1985. [cited at p. 8, 15]

[67] 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. [cited at p. 1]

[68] 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. [cited at p. 1, 2, 11, 17, 23]

[69] M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28(2):253–266, 2004. [cited at p. 1, 2, 3, 23, 123]

[70] 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, *Symposium on Discrete Events and Manufacturing Systems. CESA '96 IMACS Multiconference*, pages 1–12, Lille, France, July 1996. [cited at p. 42, 43]

[71] 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. [cited at p. 1, 14, 15, 16]

[72] P. K. Sinha. *Multivariable Control: An Introduction*. Marcel Dekker, 1984. [cited at p. 54, 68, 98]

[73] R. H. Sloan and U. Buy. Reduction rules for time Petri nets. *Acta Informatica*, 33(7):687–706, 1996. [cited at p. 44]

[74] M. Sznaier and M.J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of the* 26^{st} *IEEE Conference on Decission and Control (CDC 2001)*, pages 761–762, December 1987. [cited at p. 129]

[75] E. Teruel, J. M. Colom, and M. Silva. Choice-free Petri nets: A model for deterministic concurrent systems with bulk services and arrivals. *IEEE Trans. on Systems, Man, and Cybernetics*, 27(1):73–83, 1997. [cited at p. 28, 39, 108]

[76] E. Teruel and M. Silva. Structure theory of equal conflict systems. *Theoretical Computer Science*, 153(1-2):271–300, 1996. [cited at p. 28, 33]

[77] P. Tøndel, T.A. Johansen, and A. Bemporad. An efficient algorithm for multi-parametric quadratic programming and explicit mpc solutions. In *Proceedings of the* 40^{st} *IEEE Conference on Decission and Control (CDC 2001)*, Orlando, Florida, USA, December 2001. [cited at p. 128]

[78] Kishor S. Trivedi and Vidyadhar G. Kulkarni. Fspns: Fluid stochastic petri nets. In *Application and Theory of Petri Nets*, pages 24–31, 1993. [cited at p. 1]

[79] R. Vidal, A. Chiuso, S. Soatto, and S. Sastry. Observability of linear hybrid systems. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003*, volume 2623 of *Lecture Notes in Computer Science*, pages 526–539, Prague, Czech Republic, 2003. Springer Berlin / Heidelberg. [cited at p. 55]

[80] J. Zhang. Performance study of markov modulated fluid flow models with priority traffic. In *Proc. of the IEEE INFOCOM'93*, pages 10–17, 1993. [cited at p. 1]