Técnicas Algebraicas para el Análisis y Control de Redes de Petri Continuas

Jorge Emilio Júlvez Bueno

TESIS DOCTORAL

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Directores: Manuel Silva Suárez
Laura Recalde Frisón

Noviembre 2004

Algebraic Techniques for the Analysis and Control of Continuous Petri Nets

Jorge Emilio Júlvez Bueno

TESIS DOCTORAL

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Directores: Manuel Silva Suárez
Laura Recalde Frisón

Noviembre 2004

"If Achilles races a tortoise and gives the tortoise a head start, then Achilles will never be able to overtake the tortoise. When Achilles reaches the point where the tortoise started, the tortoise will have moved on. When Achilles reaches that point, the tortoise will have moved on farther - and so on indefinitely."

Zeno of Elea. Achilles Paradox.

Acknowledgements

I am very grateful to Manuel Silva Suárez for his ideas, time and great expertise in the Petri nets world. Many thanks as well to Laura Recalde Frisón for her close collaboration and for being so involved in every matter of this work. Both have effectively guided me during my PhD and have definitively contributed to carry out the research presented in this work.

Thanks a lot to Alessandro Giua, Alberto Bemporad and René Boel for their hospitality and for the very pleasant research stays they offered me in their research groups. They helped me to focus and face Petri nets problems from different points of view. Many thanks to Daniele Corona, Emilio Jiménez and Carla Seatzu for having coauthored the works developed during this thesis.

I would like to thank the members of the Petri nets group of the Universidad de Zaragoza, specially, I am very grateful to Javier Campos for having tempted me to apply for a PhD grant four years ago. I also would like to thank all the excellent mates I have met in the research laboratory and in the DIISasters team for the fantastic atmosphere they create.

Let me also thank the institutions that provided this work with financial support: A PhD grant from the Diputación General de Aragón (reference B106/2001), the projects CICYT and FEDER DPI2003-06376 and TIC2001-1819 from the Spanish Ministerio de Educación y Ciencia, and a European Community Marie Curie Fellowship, Control Training Site (number: HPMT-CT-2001-00278).

I am infinitely grateful to my family and my friends. Doubtless, their warm support throughout this period has been the key for the success of this thesis.

Técnicas algebraicas para el análisis y control de Redes de Petri Continuas

Resumen

Las redes de Petri constituyen un potente formalismo para el modelado y análisis de sistemas concurrentes. Tradicionalmente, las redes de Petri han sido utilizadas en el contexto de sistemas discretos. Uno de los mayores problemas que aparece en sistemas discretos altamente poblados es el de la explosión de estados: El número de estados del sistema crece exponencialmente con respecto a su población inicial. La fluidificación o continuización es una técnica de relajación clásica cuyo objetivo es evitar la aparición de este problema.

Este trabajo está dedicado al estudio de las redes de Petri continuas. En una red de Petri continua el disparo de las transiciones no está restringido al conjunto de números naturales sino al de los reales positivos. De este modo, el estado/marcado de una red continua viene dado por un vector de números reales. En redes de Petri continuas el espacio de estados alcanzables es convexo lo que permite el uso de técnicas lineales en vez de enteras. Este hecho repercute muy positivamente en la complejidad de los algoritmos de verificación.

Por desgracia, la red de Petri fluidificada no siempre preserva las propiedades de la red discreta original. Por ejemplo la vivacidad de la red discreta no es una condición suficiente ni necesaria para la vivacidad de la red fluidificada. Esta y otras discrepancias entre las redes discretas y sus fluidificadas dan a entender que las redes de Petri continuas requieren un estudio independiente y riguroso.

El presente documento trata tanto redes continuas no temporizadas como temporizadas. Las principales propiedades que se estudian en redes no temporizadas son alcanzabilidad y vivacidad. Con respecto a redes temporizadas los temas investigados están relacionados con vivacidad, evaluación del rendimiento, observabilidad y controlabilidad.

Contents

Introduction

Discrete systems with large populations or heavy traffic appear frequently in many fields: manufacturing processes, logistics, telecommunication systems, traffic systems,... It becomes therefore interesting to develop adequate formalisms and tools for the analysis and verification of highly populated systems. In principle, the "natural" approach to study such systems is through the use of discrete models. Several formalisms as max-plus algebras, Markov processes and Petri nets have been proposed to model and analyze the behaviour of dynamical discrete event systems. The popularity of Petri nets [Pet81, Bra83, Sil85, Mur89, Sil93] is due to several reasons: On the one hand it offers an intuitive graphical format based on very few simple primitives that makes modelling tasks relatively easy. On the other hand, this reduced number of primitives allows one to model a wide variety of system behaviours as concurrency, synchronization, competition, cooperation, etc.

One of the main drawbacks inherent to discrete event systems is that they suffer from the state explosion problem. This phenomenon leads to an exponential growth of the size of the state space with respect to the size of the system. The undesirable state explosion makes that some system properties are computationally too heavy to be checked if an exhaustive exploration of the state space is required. In the framework of Petri nets, structural techniques [Sif78, SC95] have been successfully developed to avoid a comprehensive enumeration of states for the verification of some properties. An interesting advantage of using structural techniques is that the results they yield apply to any initial state of the system. Unfortunately, they often offer only semidecision conditions, i.e., either necessary or sufficient conditions, or their application is restricted to some net subclasses.

A way to face the state explosion problem is to relax the original discrete model in order to deal with an "equivalent", more friendly, non-discrete model. Fluidification is a classical relaxation technique whose goal is to transform a discrete system into a continuous system with similar properties and behaviours. The subject of study of this thesis is continuous Petri nets, i.e., Petri nets to which fluidification has been applied in order to avoid the state explosion problem.

In a continuous Petri net the firing of a transition is not constrained to the natural

numbers but to the nonnegative real numbers. Thus, when a transition is fired, a real (not necessarily natural) amount of tokens is removed from the input places of the transition and a real amount of tokens 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 a continuous Petri net transitions can be seen as valves through which "fluid tokens" flow, and 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 any more but a continuous region.

The fluidification of discrete Petri nets does not only avoid the state explosion problem but also gives one the chance of using linear programming techniques instead of integer programming techniques. This fact clearly involves a great computational gain since linear programming problems can be solved in polynomial time while integer programming problems usually entail an exponential complexity.

Continuous Petri nets inherit many interesting concepts of discrete Petri nets. In particular, the concepts based on the representation of the net as a graph can be directly applied to continuous nets. For example, the conflict relationships among transitions and the concepts of (P-)T-semiflows, siphons and traps can be directly applied to continuous Petri nets. However, the behaviour of a continuous Petri net system with respect to these concepts is not necessarily equivalent to that of the original discrete net system. This non equivalent behaviour appears, for instance, when considering a trap: In contrast to a trap in a discrete Petri net, a marked trap in a continuous Petri net might be emptied if infinitely long firing sequences are allowed [Rec98]. Thus, a basic behavioural property of the discrete system as the impossibility for a marked trap to become empty could be violated by the fluidified system. This fact can be interpreted as if some discrete net systems could not be reasonably fluidified.

Unfortunately, the number of behavioural discrepancies between discrete net systems and the fluidified versions is substantial and cannot be overlooked. If one considers qualitative properties, it is remarkable that a crucial system property as liveness is not in general preserved by the fluidified net system. With respect to quantitative properties, it could be thought that since the firing of transitions is not restricted to the natural numbers, the performance of the fluidified net system should be an upper bound for the performance of the original discrete system. This is, however, not always the case, i.e., there exist systems that perform better as discrete than as continuous. All these phenomena and unexpected behaviours make clear that the results obtained for a continuous net system cannot be always extrapolated to the original discrete net system. In other words, the at first glance naive fluidification of Petri nets requires a thorough study. This work represents an effort to analyze and better understand the behaviour and properties of continuous Petri nets.

As in discrete Petri net systems, continuous Petri net systems can be studied without time interpretation. These net systems will be called untimed. The order and amount in which the transitions of untimed systems are fired is, in principle, non determined. In fact, as in discrete nets, the amount in which a continuous transition can be fired is just upper bounded by its enabling degree. In a continuous net system, the set of reachable markings is the result of considering the net markings obtained by all possible firing sequences. In addition to this set of reachable markings, Chapter 2 shows that the concept of reachability can be refined in two ways: by considering as reachable those markings that can be reached by firing an infinitely long sequence; or by considering as reachable those markings to which the system can get as close as desired. It can be proved that there exists an inclusion relationship among all three reachability sets. Furthermore, the three reachability sets are very similar: The differences (if any) among all three sets lay only in the border points of the reachability sets. It turns out that the reachability set (under any reachability concept) of any continuous Petri net is a convex set and can be fully characterized by using the fundamental state equation and some other mathematical conditions. This way, the problem of checking wether a given marking is reachable in a continuous Petri net is decidable under any reachability concept. Therefore, in continuous Petri nets the existence/absence of non desired potentially reachable markings can be checked without the use of reachability trees.

One of the most often requested system properties is deadlock-freeness. A net system is deadlock-free when it is impossible to reach a marking from which no transition can be fired, in other words, from any reachable marking there is a transition that can be fired. A stronger condition than deadlock-freeness is liveness: A system is live iff for any transition, t, and for any reachable marking, m, there exists a fireable sequence from m that fires transition t. It derives that deadlock-freeness is a necessary condition for liveness. Chapter 3 is devoted to the study of deadlock-freeness for the subclass of untimed continuous mono-T-semiflow Petri nets, i.e., nets that are conservative, consistent and have only one T-semiflow. For this subclass of nets deadlock-freeness and liveness are equivalent. In order to be live, the transitions of a mono-T-semiflow Petri net have to be fired according to the ratios given by the unique T-semiflow. This fact allows one to extract easy to check structural conditions for liveness.

Time can be introduced in the continuous Petri net formalism in several ways. It is more common and intuitive to associate time to transitions than to places. The most popular time interpretations for continuous transitions are finite servers semantics [AD98a] (or constant speed) and infinite servers semantics [RS01] (or variable speed). Both firing semantics are derived by considering a first order approximation of the discrete case. The firing semantics define the flow (number of firings per time unit) through transitions. Once the flow through transitions is known, the evolution

of the marking can be computed by using the fundamental state equation.

Under finite servers semantics the flow of a transition keeps constant as long as none of its input places becomes empty. If one of the input places becomes empty the flow of the transition changes to a value that depends on the flow of the transition that is providing fluid to the empty place. Thus, the evolution of the marking is piecewise linear: A change in the marking dynamics happens when a place becomes empty. Finite servers semantics are suitable to model systems whose queues/warehouses are served at constant rates.

The flow of a transition working under infinite servers semantics is proportional to its enabling degree. The enabling degree of a transition depends on the marking of its input places and the weight of the arcs connecting the input places to the transition. More precisely, the enabling degree is computed by considering the division of the marking of each input place by its arc weight and taking the minimum of those divisions. The place that is giving the minimum division is somehow constraining the firing of the transition. As under finite servers semantics, the evolution of a net system under infinite servers semantics is piecewise linear. Now, a change in the marking dynamics happens when there is a change in the place giving the minimum in the expression for the enabling degree of a given transition.

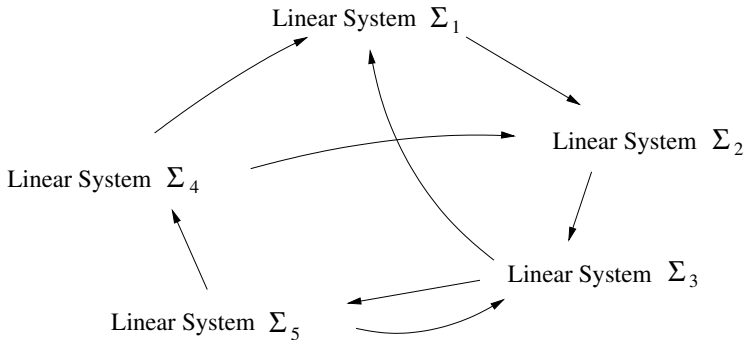


Figure 1: Timed continuous Petri net system as piecewise linear system. Each arc corresponds to an internal event.

Thus, the differences between finite and infinite servers semantics are only in the kind of internal events triggering the dynamics change and the kind of linear systems driving the evolution of the marking: Under finite servers semantics an event occurs when a place becomes empty and the flow through transitions is constant at every linear system. Under infinite servers semantics an event is activated by a change in the place giving the minimum for the enabling degree of a transition and the flow through transitions is proportional to their enabling degree.

The properties of deadlock-freeness/liveness can be studied as well in the frame-

work of timed continuous net systems. A net system is said to be live if the flow of all its transitions is greater than zero in the steady state. As in untimed systems, in timed mono-T-semiflow net systems deadlock-freeness and liveness are equivalent. Chapter 4 shows how some liveness conditions for timed mono-T-semiflow nets under infinite servers semantics can be obtained. Notice that the marking trajectory of the timed net system is contained in the reachability set of the net system seen as untimed. This implies that if the timed system deadlocks it can also deadlock as untimed. In other words, liveness of the timed system is a necessary condition for liveness of the untimed system. This relationship between liveness of timed and untimed systems can be exploited to extract necessary liveness conditions for untimed systems.

The performance of a dynamical system is a real quantity measuring how well a system behaves. Typically, the higher the performance the better the behaviour. The performance of a system can be obtained for example by computing how many times a given action is executed per time unit. In *discrete* Petri net systems, the performance of a system is computed as the number of times a given transition is fired per time unit. In a similar way, in continuous Petri net systems the performance is measured as the flow (or throughput) of a transition at the steady state, i.e., when the marking and flow through transitions remain constant. In mono-T-semiflow systems the only sequences that can be indefinitely repeated have to be proportional to the unique T-semiflow. Therefore, the vector representing the flow through transitions at the steady state has to be proportional to the unique T-semiflow of the net. Thus, once the throughput of one transition in the steady state is known, the throughput of the rest of transitions can be immediately obtained. A Branch & Bound algorithm can be used to compute throughput bounds of a net system. By slightly relaxing some conditions it is also possible to derive a linear programming problem for the computation of upper throughput bounds. These methods are presented in Chapter 5.

In many real situations the state variables of a dynamical system are not fully accessible, i.e., they cannot be directly measured by an external observer. This can be the case of internal system variables and variables for which sensors do not exist or are too expensive. Fortunately, under some conditions, it is possible to estimate/observe the value of those "hidden" variables. In the framework of linear systems, it is relatively easy to obtain the observability space of the system, i.e., the state space that can be estimated from the output (the measurable variables) of the system. It is also possible to design, under some conditions, a dynamical system called *observer* that estimates the value of the hidden variables. The observer's input is the output of the system to be observed and its state gives an estimate for the hidden variables. Some results on observability for linear systems and piecewise linear systems can be applied to timed continuous Petri nets. Furthermore, Chapter 6 shows that these results can be improved by considering some specific features of continuous nets. Basically, this improvement is possible thanks to the fact that the linear system that rules the net

system evolution depends exclusively on the marking of the net.

Controllability is the dual property of observability. Intuitively, a system is said to be controllable when its state can be completely manipulated by means of input actions. In Petri nets, input actions can be added to the model by introducing the possibility of modifying the "normal unforced" flow of the transitions, i.e., its flow according to finite or infinite servers semantics. The unforced flow of a transition can be seen as its maximum working speed since it works completely unconstrained. An input action is applied to a transition in order to slow down its normal working speed. Following these ideas Chapter 6 proposes a method to solve optimal control problems in timed continuous Petri nets. These problems aim to maximize/minimize a given optimization function subject to the rules of the evolution of the net system.

Chapter 8 studies three different systems modelled with continuous Petri nets: A manufacturing system, an assembly line and a car traffic system. The two first cases strongly suffer from the state explosion problem. It is shown how they can be analyzed as continuous models by using the concepts developed in this thesis. The third case represents an effort to faithfully model a car traffic system. For that purpose some model extensions, that go beyond the usual definitions for continuous Petri nets, are proposed.

Chapter 1

Continuous Petri Nets

Summary

This chapter introduces some basic definitions and concepts related to discrete and continuous Petri nets. Some Petri net subclasses are presented allowing one to classify a net according to its structure. In particular, the subclass of mono-T-semiflow nets is described more deeply. This subclass has interesting analysis features that will be studied in the following chapters. A continuous Petri net is presented as a relaxation of a discrete one. In a continuous Petri net the marking of a place and the firing of a transition are not discrete amounts but nonnegative real amounts. This relaxation may not preserve some basic properties of the original discrete system. Two different possibilities to introduce time in a continuous Petri net system are described: Finite servers semantics and infinite servers semantics. Under any of these two time interpretations, timed Petri net systems are particular cases of piecewise linear systems.

Introduction

Petri nets are widely used to model, analyze and verify discrete event systems. One of the reasons for the success of Petri nets is that many behaviours of discrete systems as concurrence, synchronization, mutual exclusion, resource sharing and coordination can be modelled in a compact and intuitive way by Petri nets. Besides, there exist many analysis techniques and methods for the verification of systems modelled by Petri nets.

One of the main drawbacks of discrete Petri nets is that they suffer from the state explosion problem, i.e., the number of states increases exponentially with respect to the size of the system. That problem affects many discrete models causing some verification methods to be computationally very expensive. A way to avoid the state explosion problem is to relax the model. A classical relaxation technique is to fluidify the discrete model in order to obtain an "equivalent" continuous one. A fluidified model gives one the chance of using linear methods instead of integer methods to analyze the model. Usually, the complexity of linear methods is polynomial in contrast to the exponential complexity of many integer methods. Unfortunately, some properties, as liveness, of the original discrete model may not be verified by the fluidified one.

The basic concepts and notations of discrete Petri nets are introduced in Section 1.1. Based on structural criteria several net subclasses are defined. Untimed continuous Petri nets are presented in Section 1.2. The firing rule of the continuous transitions is somehow similar to the rule for discrete transitions. However, the transitions can now be fired in nonnegative real amounts. This causes the marking of the net to be a vector of nonnegative real numbers. The concept of time is introduced in Section 1.3 as a first order approximation of the discrete case. Two different firing semantics are considered: Finite servers semantics and infinite servers semantics. Under finite servers semantics the flow of a transition depends on the set of its empty input places. On the other hand, under infinite servers semantics the flow is proportional to the enabling degree of the transition. In both cases, finite and infinite servers semantics, the evolution of a timed continuous Petri net follows the pattern of a piecewise linear system. Finally, Section 1.4 outlines some behaviours of the continuized Petri net system that are not as coherent with the behaviour of the original discrete system as it could be expected. In particular, liveness is not preserved and the performance of the continuized system is not an upper bound of the performance of the original discrete system. These facts motivate a detailed study of continuous Petri nets.

1.1 *Discrete* Petri nets and systems. State explosion problem

The reader is assumed to be familiar with the basic concepts of discrete Petri nets (see [Pet81, Bra83, Sil85, Mur89, Sil93] for an introduction). A *Petri net* (PN) is a four-tuple, $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, where P and T are disjoint (finite) sets of *places* and *transitions*, and **Pre** and **Post** are $|P| \times |T|$ sized, non-negative integer valued matrices. When all weights are one the net is *ordinary*.

A PN can be graphically represented as a weighted bipartite directed graph: Places are drawn as circles and transitions as white rectangles, $\mathbf{Pre}[p,t] = w > 0$ means that there is an *arc* from p to t with *weight* (or multiplicity) w, and $\mathbf{Post}[p,t] = w > 0$ means that there is an arc from t to p with weight w. Thus, classical concepts of graph theory, as connectedness, strong connectedness, adjacent nodes,..., can be directly applied to PN nets. Given a node $v \in P \cup T$, its *preset*, ${}^\bullet v$, is defined as the set of its input nodes, and its *postset* $v^\bullet$ as the set of its output nodes. These definitions can be naturally extended to sets of nodes.

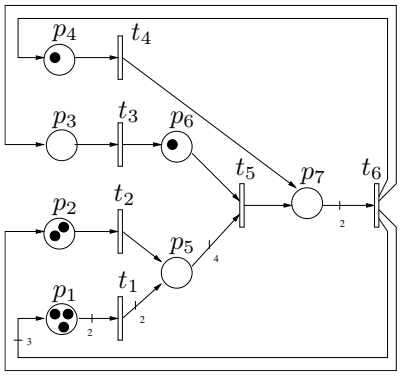
If no place is at the same time input and output of a transition, i.e. $\forall t \in T \;\; {}^\bullet t \cap t^\bullet = \emptyset$, the net has no self-loops and is *pure*. In this case the *token flow matrix*, $\mathbf{C} = \mathbf{Pre} - \mathbf{Post}$, contains all the information of the **Pre** and **Post** matrices and it is also called incidence matrix.

A *marking* is a $|P|$ sized, natural valued, vector. A *PN system* is a pair $\langle \mathcal{N}, \mathbf{m_0} \rangle$, where $\mathbf{m_0}$ is called the initial marking, i.e., the initial state of the system.

A transition t is *enabled* at a marking $\mathbf{m}$ iff $\mathbf{m} \geq \mathbf{Pre}[P,t]$. The firing of an enabled transition t produces a new marking $\mathbf{m'} = \mathbf{m} + \mathbf{C}[P,t]$. The firing is denoted by $\mathbf{m} \xrightarrow{t} \mathbf{m'}$, and $\mathbf{m'}$ is said to be a reachable marking (from $\mathbf{m}$). The set of all the reachable markings, or *reachability set*, from $\mathbf{m}$, is denoted by $\mathrm{RS}(\mathcal{N}, \mathbf{m})$.

The size of the reachability set of a PN system can increase exponentially with respect to the initial marking. Figure 1.1 presents a PN system and a table showing how the size of the reachability set increases as the initial marking $\mathbf{m_0} = (3\ 2\ 0\ 1\ 0\ 1\ 0)$ is multiplied by a constant k. This phenomenon is known as the *state explosion problem* and poses serious difficulties when an exhaustive exploration of the reachability set is required.

Given a sequence of transitions σ such that $\mathbf{m} \xrightarrow{\sigma} \mathbf{m'}$, and denoting by $\boldsymbol{\sigma}$ the *firing count vector* of σ, then $\mathbf{m'} = \mathbf{m} + \mathbf{C} \cdot \boldsymbol{\sigma}$. This is known as the *state (or fundamental) equation* of the system. The set of all the markings that fulfill the state equation for a given $\mathbf{m} \in \mathbb{N}^{|P|}$, with $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}$, is called the *linearized reachability set* (with respect to the state equation), $\mathrm{LRS}(\mathcal{N})$.



k	Number of reachable markings
1	54
2	1685
3	10354
4	37722
5	103914

Figure 1.1: A discrete Petri net and the size of its reachability set.

1.1.1 Some basic concepts

Flows and semiflows

Flows (*semiflows*) are integer (natural) annullers of $\mathbf{C}$. Right and left annullers are called T- and P-(semi)flows, respectively. A semiflow $\mathbf{v}$ is *minimal* when its support, $\|\mathbf{v}\| = \{i \mid \mathbf{v}[i] \neq 0\}$, is not a proper superset of the support of any other semiflow, and the greatest common divisor of its elements is one.

If there exists $\mathbf{y} > \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$ then the net is said to be *conservative* and it holds:

$$\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m_0} + \mathbf{y} \cdot \mathbf{C} \cdot \boldsymbol{\sigma} = \mathbf{y} \cdot \mathbf{m_0} = k$$

This provides a token balance law for the whole net. In a similar way, if there exists $\mathbf{x} > \mathbf{0}$ such that $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$ then the net is said to be *consistent* and it holds:

$$\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \mathbf{x} = \mathbf{m_0}$$

Hence, $\mathbf{x}$ represents a *potential* repetitive sequence covering all transitions.

Traps and siphons

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 marked traps cannot get emptied. Analogously, a set of places, Φ, is a *siphon* iff ${}^\bullet\Phi \subseteq \Phi^\bullet$. One interesting property about siphons is that empty siphons will unavoidably remain empty throughout all the evolution of the net system.

Conflicts

A conflict is the situation where not all transitions that are enabled can occur at the same time. More formally, $t, t' \in T$ are in (effective) conflict relation at marking $\mathbf{m}$ iff there exist $k, k' \in \mathbb{N}$ such that $\mathbf{m} \geq k \cdot \mathbf{Pre}[P, t]$ and $\mathbf{m} \geq k' \cdot \mathbf{Pre}[P, t']$, but $\mathbf{m} \not\geq k \cdot \mathbf{Pre}[P, t] + k' \cdot \mathbf{Pre}[P, t']$. For this, it is necessary that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, and in that case it is said that t and t' are in structural conflict relation. The structural conflict relation (or *choice*) is a structural prerequisite for the behavioural property of conflict.

The structural conflict relation is not transitive, and the *coupled conflict* relation is defined as its transitive closure. Each equivalence class is called a coupled conflict set denoted, for a given t, CCS(t). The set of all the equivalence classes is denoted by SCCS.

When $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t'] \neq \mathbf{0}$, t and t' are in *equal conflict* (EQ) relation, meaning that they are both enabled whenever one is. This is an equivalence relation on the set of transitions and each equivalence class is an equal conflict set denoted, for a given t, EQS(t). An equal conflict set is called trivial if it is formed by just one transition. SEQS is the set of all the equal conflict sets of a given net.

Liveness and deadlock-freeness

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

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

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

1.1.2 Petri net subclasses

Typically, Petri net subclasses are defined by imposing some constraints on the structure of the net. The following ones are among the most usual net subclasses:

Definition 1.1 (Some Petri net subclasses).

- State machines (SM) are ordinary Petri nets where each transition has one input and one output place, i.e., $\forall t \ \ |{}^\bullet t| = |t^\bullet| = 1$.
- Marked graphs (MG) [CHEP71] are ordinary Petri nets where each place has one input and one output transition, i.e., $\forall p \ \ |{}^\bullet p| = |p^\bullet| = 1$.

- Join free (JF) nets are Petri nets in which each transition has at most one input place, i.e., $\forall t \in T$, $|{}^\bullet t| \leq 1$).
- Choice free (CF) nets [TCS97] are Petri nets in which each place has at most one output transition, i.e., $\forall p \ \ |p^\bullet| \leq 1$.
- Free choice (FC) nets [Hac72] are ordinary Petri nets in which conflicts are always equal, i.e., $\forall t, t'$, if ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, then ${}^\bullet t = {}^\bullet t'$.
- Equal Conflict (EQ) nets [TS96] are Petri nets in which conflicts are always equal, i.e., for all $t, t' \in T$ such that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t']$, or, equivalently, SCCS = SEQS.

Mono-T-semiflow nets

A Petri net is Mono-T-semiflow (MTS)[CCS91] if it is conservative (i.e., all places are covered by P-semiflows), consistent and has only one T-semiflow (i.e., all transitions are covered by *the* unique minimal T-semiflow). Hence, it can be decided in polynomial time whether a given net, $\mathcal{N}$, is MTS or not.

MTS nets have a single repetitive sequence that is represented by its unique T-semiflow. This offers interesting analysis advantages and implies the equivalence between deadlock-freeness and liveness. From a modelling point of view the subclass of MTS nets represents an important generalization of *choice-free* nets (see Figure 1.2). A subclass of choice free nets are weighted-T-systems [TCS97], a weighted generalization of the subclass of marked graphs.

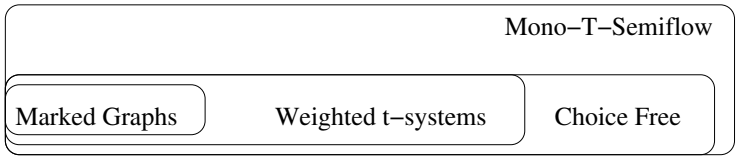


Figure 1.2: Subclasses included in the subclass of MTS nets.

1.2 Untimed continuous Petri net systems

An interesting approach to study discrete systems with large populations is based on the fluidification/continuization of the model. Thus, the model is not discrete any more but continuous. This is a relaxation technique that can also be applied in the context of Petri nets in order to overcome the state explosion problem. Usually, but not always [SR02], the greater the population of the discrete system the better the continuous approximation.

In PNs, fluidification has been introduced independently from three different perspectives:

- At the net level fluidification was introduced and developed by R. David and coauthors since 1987 [DA87, AD98a]. In this case, the fluidification of timed discrete systems generates deterministic continuous models, and also hybrid models if there is a partial fluidification.
- Analogously, fluidifying the firing count vector (thus also the marking) in the state equation allows one to use convex geometry and linear programming instead of integer programming, making possible the verification of some properties in polynomial time. The systematic use of linear programming on untimed and timed systems was proposed also in 1987 [SC88, STC98].
- K. Trivedi and his group introduced [TK93, CNT99] a partial fluidification on some stochastic models. The fluidification only affects one or a limited number of places originating stochastic hybrid systems.

In this work, a continuous system is understood as a relaxation of a *discrete* system. The main difference between continuous and discrete PNs is in the firing count vector and consequently in the marking, which in discrete PNs are restricted to be in the naturals, while in continuous PNs are relaxed into the non-negative real numbers. The marking of a place can be seen as an amount of fluid being stored, and the firing of a transition can be considered as a flow of this fluid going from a set of places (input places) to another set of places (output places). Thus, instead of tokens and discrete firings, it is more convenient to talk of levels in the places (deposits/reservoirs) and flows through transitions (valves).

A transition t is *enabled* at $\mathbf{m}$ iff for every $p \in {}^\bullet t$, $\mathbf{m}[p] > 0$. In other words, the enabling condition of continuous systems and that of discrete ordinary systems can be expressed in an "analogous" way: every input place should be marked. Notice that to decide whether a transition in a continuous system is enabled or not it is not necessary to consider the weights of the arcs going from the input places to the transition. However, the arc weights are important to compute the enabling degree of a transition and to obtain the new marking after a firing. As in discrete systems, the *enabling degree* at $\mathbf{m}$ of a transition measures the maximal amount in which the transition can be fired in a single occurrence, i.e., $\mathrm{enab}(t, \mathbf{m}) = \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p, t]\}$. The firing of t in a certain amount $\alpha \leq \mathrm{enab}(t, \mathbf{m})$ leads to a new marking $\mathbf{m}'$, and it is denoted as $\mathbf{m} \xrightarrow{\alpha t} \mathbf{m}'$. It holds $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}[P, t]$, where $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the token flow matrix. Hence, as in discrete systems, the state equation, $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$, summarizes the way the marking evolves.

As in discrete systems, when $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$, $\mathbf{y} > \mathbf{0}$ the net is said to be *conservative*, and when $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$, $\mathbf{x} > \mathbf{0}$ the net is said to be *consistent*. As an immediate generalization

of equal conflict relations, it will be said that t and t' are in *continuous equal conflict (CEQ)* relation when there exists $k > 0$ such that $\mathbf{Pre}[P, t] = k \cdot \mathbf{Pre}[P, t'] \neq \mathbf{0}$.

In order to illustrate the firing rule in a continuous system, let us consider the system in Figure 1.3. The only enabled transition at the initial marking is t_1 whose enabling degree is 1. Hence, it can be fired in any real quantity going from 0 to 1. For example, the firing by 0.5 would yield marking $\mathbf{m}_1 = (0.5,\ 0.5,\ 1,\ 0)$. At $\mathbf{m}_1$ the enabling degree of transition t_2 is equal to 0.5; if it is fired in this amount the resulting marking is $\mathbf{m}_2 = (0.5,\ 0.5,\ 0,\ 0.5)$. Both $\mathbf{m}_1$ and $\mathbf{m}_2$ are reachable markings of the continuous Petri net system.

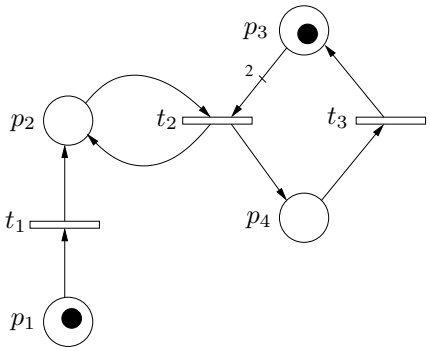


Figure 1.3: Untimed continuous Petri net system.

1.3 Timed continuous Petri net systems

For the timing interpretation of *continuous* PNs a first order (or deterministic) approximation of the discrete case [RS01] will be used, assuming that the delays associated to the firing of transitions can be approximated by their mean values. Then, the state equation has an explicit dependence on time $\mathbf{m}(\tau) = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}(\tau)$. Deriving with respect to time, $\dot{\mathbf{m}}(\tau) = \mathbf{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$ is obtained. Let us denote $\mathbf{f} = \dot{\boldsymbol{\sigma}}$, since it represents the flow of the transitions.

Different semantics have been defined for continuous PNs, the most important being *finite servers* [AD98a](or constant speed) and *infinite servers* [RS01](or variable speed).

1.3.1 Finite servers semantics

Under finite firing semantics, every transition, t, has associated a real parameter $\boldsymbol{\lambda}[t] > 0$ that is the maximum flow of the transition. Intuitively, if a transition is seen as a valve through which a fluid passes, $\boldsymbol{\lambda}$ can be seen as the maximum flow admitted

by the valve. In contrast to [BGM00] no lower bound for the flow of the transitions will be considered along this work, thus, the minimum flow of every transition is 0. A transition, t, is *strongly enabled* if every input place of t is marked or t has no input places. If t is strongly enabled then $\mathbf{f}[t] = \boldsymbol{\lambda}[t]$. A transition, t, is *weakly enabled* if one or more input places of t are empty and receiving an input flow from some of their input transitions, and the rest of input places of t are marked. The flow of the weakly enabled transitions has to be defined in such a way that the nonnegativity of the marking is assured. The computation of an admissible $\mathbf{f}$ is not trivial when several empty places appear. In [AD98b], an iterative algorithm is suggested to compute one admissible $\mathbf{f}$. In this work, $\mathbf{f}$ will be computed in a similar way to [BGM00] where the set of admissible $\mathbf{f}$ is characterized by a set of linear inequalities. The vector $\mathbf{f}$ that will be chosen fulfils the system of linear inequalities and maximizes $\sum_{i=1}^{s} \mathbf{f}[t_i]$. If a transition is neither strongly nor weakly enabled its flow is 0.

Under finite server semantics, the flow vector, $\mathbf{f}$, is piecewise constant, and therefore the marking evolution is piecewise linear. The vector $\mathbf{f}$ keeps constant until an event occurs. Between events, the system is said to be at an *invariant behavior state* (IB - state) [AD98a]. An event occurs only when a place becomes empty. Thus, the number of potential IB - states equals the number of sets of places that can be empty. In principle, each place can be empty or not empty, hence the number of potential IB - states for a general system with n places is 2^n. However, the number of potential IB - states is usually not so big since initially marked P-semiflows ([Mur89, DHP+93]) cannot be emptied.

Let us consider the system in Figure 1.4(a). The only input place of t_1 is marked, hence it is strongly enabled and $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] = 2$. The evolution of $\mathbf{m}[p_1]$ is given by $\dot{\mathbf{m}} = \boldsymbol{\lambda}[t_2] - \boldsymbol{\lambda}[t_1] = -1$. At time 1, p_1 becomes empty, i.e., an event occurs, and t_1 becomes weakly enabled. Now, the maximum flow admitted by t_1 is 1, a greater flow will cause $\mathbf{m}[p_1]$ to be negative. Being $\mathbf{f}[t_1] = 1$, p_1 remains empty. Now p_1 can be seen as a tube instead of a deposit and no more events occur. For arbitrary values of $\boldsymbol{\lambda}[t_1]$ and $\boldsymbol{\lambda}[t_2]$, the flow of t_1 when p_1 is empty is defined as $\mathbf{f}[t_1] = \min(\boldsymbol{\lambda}[t_1], \boldsymbol{\lambda}[t_2])$.

Transitions t_1 and t_2 of the system in Figure 1.4(b) are strongly enabled for the given initial marking $\mathbf{m}[p_1] = 2$. After two time units, p_1 becomes empty and t_1, t_2 become weakly enabled. At this point, it has to be decided how to split the input flow, $\boldsymbol{\lambda}[t_3]$, coming into p_1. Since t_1 and t_2 are considered to have the same priority, half of the flow should be routed to t_1 and half to t_2. Unfortunately, the maximum flow of t_1, $\boldsymbol{\lambda}[t_1] = 1$, is smaller than $\boldsymbol{\lambda}[t_3]/2$. To solve this situation, the flow that cannot be consumed by t_1, $\boldsymbol{\lambda}[t_3]/2 - \boldsymbol{\lambda}[t_1]$, is routed to t_2. This results in $\mathbf{f}[t_1] = 1$ and $\mathbf{f}[t_2] = 2$. The explicit analytic expressions for $\mathbf{f}[t_1]$ and $\mathbf{f}[t_2]$ with arbitrary $\boldsymbol{\lambda}[t_1]$, $\boldsymbol{\lambda}[t_2]$ and $\boldsymbol{\lambda}[t_3]$ when $\mathbf{m}[p_1] = 0$ are $\mathbf{f}[t_1] = \min(\boldsymbol{\lambda}[t_3]/2 + \max(0, \boldsymbol{\lambda}[t_3]/2 - \boldsymbol{\lambda}[t_2]), \boldsymbol{\lambda}[t_1])$ and $\mathbf{f}[t_2] = \min(\boldsymbol{\lambda}[t_3]/2 + \max(0, \boldsymbol{\lambda}[t_3]/2 - \boldsymbol{\lambda}[t_1]), \boldsymbol{\lambda}[t_2])$.

Consider the system in Figure 1.5(a) with $\boldsymbol{\lambda} = (1.5\ 1\ 2)$ and $\mathbf{m_0} = (0\ 0\ 3)$. The

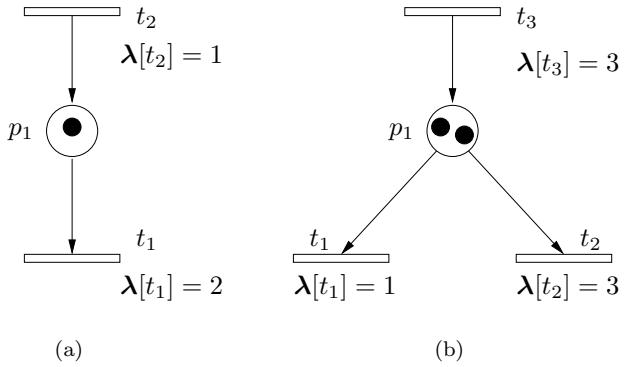


Figure 1.4: (a) Transition t_1 becomes weakly enabled at $\tau = 1$. (b) Transitions t_1 and t_2 become weakly enabled at $\tau = 2$.

evolution of the system is depicted in Figure 1.5(b). At time instant $\tau = 3$ place p_3 empties and transition t_3 becomes weakly enabled, and so its flow changes. At time $\tau = 12$ place p_1 also becomes empty and the system dies, that is, the flow of every transition is zero.

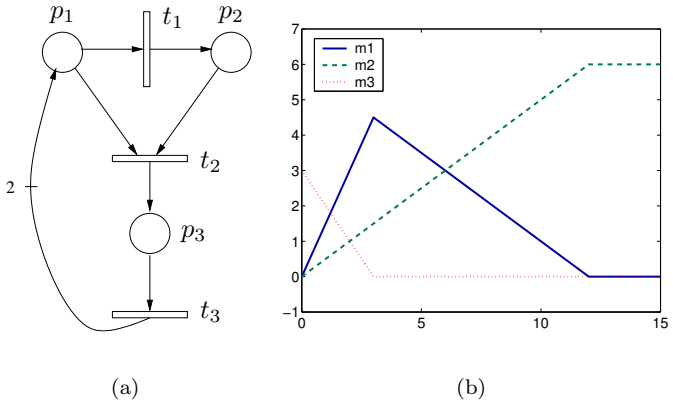


Figure 1.5: A continuous PN system and its marking evolution.

1.3.2 Infinite servers semantics

Under infinite servers semantics the flow of a transition is given by the product of a parameter $\boldsymbol{\lambda}$ by its enabling degree, i.e., $\mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \text{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot$

$\min_{p\in{}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p,t]\}$, what leads to a non-linear system. The vector $\boldsymbol{\lambda}$ is constant and can be seen as the internal speed of the transitions.

In JF systems, transitions have only one input place, and so the computation of the enabling degrees does not require the *min* operator. Hence, the flow of the transitions can be expressed as $\mathbf{f} = \boldsymbol{\Psi}\cdot\mathbf{m}$ where $\boldsymbol{\Psi}[t,p] = \boldsymbol{\lambda}[t]/\mathbf{Pre}[p,t]$ if $p = {}^\bullet t$, $\boldsymbol{\Psi}[t,p] = 0$ otherwise. Consequently, the evolution of the marking can be described by an equation in the form $\dot{\mathbf{m}} = \mathbf{C}\cdot\mathbf{f} = \mathbf{A}\cdot\mathbf{m}$, where $\mathbf{A} = \mathbf{C}\cdot\boldsymbol{\Psi}$. Hence, a JF system evolves as a linear system.

For a general system, matrix $\mathbf{A}$ is not constant but piecewise-constant. The value of $\mathbf{A}$ at a given instant is determined by the marking $\mathbf{m}$ at that instant. To compute $\mathbf{A}$, it is necessary to know the set of places that is actually enabling the transitions, i.e., the set of places that are giving the minimum in the expression for the enabling degree. Once this set is computed, it is easy to establish a linear relationship between the marking of the places in this set and the flow of the transitions: $\dot{\mathbf{m}} = \mathbf{A}\cdot\mathbf{m}$, with $\mathbf{A} = \mathbf{C}\cdot\boldsymbol{\Psi}$ where

$$\boldsymbol{\Psi}[t,p] = \begin{cases} \boldsymbol{\lambda}[t]/\mathbf{Pre}[p,t] & \text{if } p\in{}^\bullet t \text{ and } \mathbf{m}[p]/\mathbf{Pre}[p,t] = \min_{q\in{}^\bullet t}\{\mathbf{m}[q]/\mathbf{Pre}[q,t]\} \\ \boldsymbol{\Psi}[t,p] = 0 & \text{otherwise} \end{cases}$$

The marking of the places restricts the behaviour/flow of their output transitions. For each marking $\mathbf{m}$, its PT-set can be defined as the set of all the pairs, (p,t), such that the marking of p is restricting the flow of transition t at marking $\mathbf{m}$.

Definition 1.2. Given a net system, the *PT-set* at a marking $\mathbf{m}$ is

$$\text{PT-set}(\mathbf{m}) = \{(p,t) \mid \mathbf{f}[t] = \boldsymbol{\lambda}[t]\cdot\mathbf{m}[p]/\mathbf{Pre}[p,t]\} \tag{1.0}$$

Obviously, for JF systems a unique PT-set exists, and $\dot{\mathbf{m}} = \mathbf{A}\cdot\mathbf{m}$. Otherwise, if the PT-set is known, the system evolves according to $\dot{\mathbf{m}} = \mathbf{A}_1\cdot\mathbf{m}$ where $\mathbf{A}_1$ depends on PT-set($\mathbf{m}$) and the $\boldsymbol{\lambda}$ of the transitions. If at a given instant the PT-set changes, i.e., a transition is restricted by other input place, the system will be ruled by another linear system $\dot{\mathbf{m}} = \mathbf{A}_2\cdot\mathbf{m}$. That is, every PT-set, k, has associated a square matrix $\mathbf{A}_k$ and a linear system $\Sigma_k : \dot{\mathbf{m}} = \mathbf{A}_k\cdot\mathbf{m}$. The set of PT-sets that will be active during the evolution of the system, i.e., *behavioral PT-sets*, depends on the net structure and the initial marking. If the initial marking is not known, the net structure defines the set of potential PT-sets, i.e., *structural PT-sets*, that might be active. Clearly, the set of structural PT-sets contains the set of behavioral PT-sets.

This way, a continuous Petri net system can be seen as a piecewise linear system [Son81] in which the switches among the linear systems are activated by internal events, i.e., the change from one PT-set to another does not need any external agent, just a certain change in the system marking. Due to the way in which the system

evolution is defined, it can be assured that the marking of the system and its first derivative with respect to time are continuous.

In order to illustrate the evolution of a non JF system, let us consider the system in Figure 1.5(a) with initial marking $\mathbf{m_0} = (3\ 0\ 0)$ and transition speeds $\boldsymbol{\lambda} = (0.9\ 1\ 1)$. If $\mathbf{m}[p_1] \leq \mathbf{m}[p_2]$, the flow of transition t_2 will be defined by the marking of p_1 (Σ_1) and the PT-set will be $\{(p_1, t_1), (p_1, t_2), (p_3, t_3)\}$. Similarly, if $\mathbf{m}[p_1] \geq \mathbf{m}[p_2]$ the flow of t_2 will be restricted by p_2 (Σ_2) and the PT-set will be $\{(p_1, t_1), (p_2, t_2), (p_3, t_3)\}$.

$$\Sigma_1 : \ \dot{\mathbf{m}} = \begin{pmatrix} -1.9 & 0 & 2 \\ -0.1 & 0 & 0 \\ 1.0 & 0 & -1 \end{pmatrix} \cdot \mathbf{m} \tag{1.1}$$

$$\Sigma_2 : \ \dot{\mathbf{m}} = \begin{pmatrix} -0.9 & -1 & 2 \\ 0.9 & -1 & 0 \\ 0.0 & 1 & -1 \end{pmatrix} \cdot \mathbf{m} \tag{1.2}$$

At the time instant in which $\mathbf{m}[p_1] = \mathbf{m}[p_2]$, Σ_1 and Σ_2 behave in the same way and any of them can be taken. Figure 1.6 shows the evolution of the system along time. At the beginning the system evolves according to Σ_2. Then a switch occurs and the dynamics of the system is described by Σ_1. A second switch turns the system back to Σ_2, the system stabilizes and no more switches take place.

Notice that for a given marking, the set of places that are not in the PT-set do not play any role in the evolution of the system. Mathematically this is expressed by null columns in the system matrix $\mathbf{A}_j$ corresponding to the places that are not in the PT-set. Such places can be temporarily considered as a kind of *timed-implicit* places, since the system evolution does not depend on them. However, when a switch occurs, at least one place that was acting as timed-implicit becomes member of the new PT-set. For the net system in Figure 1.5(a) with $\mathbf{m_0} = (3\ 0\ 0)$, p_2 is timed-implicit only in the period when Σ_1 is describing the system dynamics.

From a modelling point of view, it holds that for any bounded time invariant linear system there exists a continuous Petri net with identical behaviour [JJRS04a]. Therefore, any dynamic behaviour that can be modelled by a time invariant linear system can also be modelled by a timed continuous Petri net working under infinite servers semantics.

1.4 Discrepancies with the discrete case

Fluidification of discrete models is, in general, a classical relaxation technique aiming at computationally more efficient analysis techniques. Nevertheless, it should be

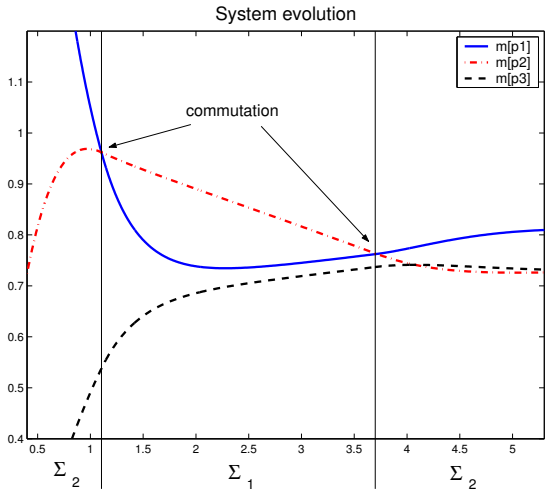


Figure 1.6: Marking evolution of the system in Figure 1.5(a) under infinite servers semantics with $\mathbf{m_0} = (3\ 0\ 0)$ and $\boldsymbol{\lambda} = (0.9\ 1\ 1)$.

pointed out that not all Petri systems allow continuization, if some basic properties as liveness should be preserved. Regarding to untimed systems, Chapter 3 presents some examples to show that liveness of the original discrete system is neither a necessary nor a sufficient condition for liveness of the continuized system. The same fact is reported for timed systems in Chapter 4.

With respect to the performance of the continuized system, it could be thought that it represents an upper bound for the performance of the original discrete system. However, in general, this is not the case. Chapter 5 focuses on the performance evaluation of continuous systems and presents a net system whose performance is higher as discrete than as continuous.

1.5 Conclusions

Continuous Petri nets represent a relaxation of classical discrete Petri nets. This relaxation avoids the state explosion problem that appears when large discrete nets are considered. In a continuous Petri net the amount in which a transition is fired is not constrained to the natural numbers but to the real numbers.

Time can be easily introduced in the continuous Petri net formalism by considering timed transitions through which a flow exists. A first order approximation of the discrete case is taken to define the flow through transitions. The flow through transitions defines the evolution of the net system. Two different firing semantics have been

described. Under finite servers (or constant speed) semantics the flow of a transition is piecewise constant and depends on the set of empty input places of the transition. This semantics is appropriate to model the behaviour of a system whose speed is not sensitive to the population (for example customers waiting to be served) of the system. On the other hand, the flow of a transition working under infinite servers semantics is proportional to its enabling degree, i.e., to the number of customers. In both cases, finite and infinite servers semantics, timed continuous Petri nets can be seen as a particular case of piecewise linear systems. Some work related to the analysis of general piecewise linear systems can be found in [Joh99, GMD03, SGL02]. In continuous Petri nets, the switch from one linear system to another one is triggered by a change in the marking of the net. This change can be interpreted as an internal event. Under finite servers semantics this event occurs when a place becomes empty. Under infinite servers semantics an event occurs when there is a change in the place defining the enabling degree for a given transition.

One could think that fluidification is a *naive* relaxation of discrete nets. However, it can cause some properties of the discrete model not to be preserved by the fluidified one. Liveness is one of those properties that can be affected by the relaxation. A thorough study of continuous Petri nets is therefore required in order to analyze their properties and to establish under which conditions the properties of the discrete model are preserved by the fluidified one.

Chapter 2

Reachability

Summary

In continuous Petri net systems reachability can be interpreted in several ways. The concepts of *reachability* and lim-*reachability* were considered in [RTS99]. They stand for those markings that can be reached with a finite and an infinite firing sequence respectively. A third concept, δ-*reachability*, can be useful for many practical purposes. A marking is δ-reachable if the system can get arbitrarily close to it with a finite firing sequence. In this chapter, a full characterization, mainly based on the state equation, is provided for all three concepts for general nets. Under the condition that every transition is fireable at least once, it holds that the state equation does not have spurious solutions if δ-reachability is considered. Furthermore, the differences among the three concepts are in the border points of the reachability sets that they define.

Introduction

The study of reachability, i.e., the set of markings that can be reached by the net system, is essential to face the analysis and verification of many system properties. For example, liveness of a system can be easily checked if a good characterization of the system reachability exists. In contrast to discrete Petri nets, in a continuous Petri net the set of reachable markings can be described by a continuous space region. This chapter shows that the use of the fundamental state equation greatly helps to describe the set of reachable markings.

Untimed Petri net models will be considered throughout this chapter. In particular, this means that no time interpretation will be applied to the firing of the transitions. Thus, a nondeterminism in the evolution of the system exists. Notice, however, that if the transitions are timed, the evolution/behaviour of the system will always be constrained to one of the possible evolutions/behaviours of the untimed system.

Three different ways of understanding (interpreting) reachability will be considered [JRS03]: reachability with a finite number of steps or simply reachability, reachability with an infinite number of steps or lim-reachability, and δ-reachability that has to do with the capacity of the system to get arbitrarily close to a given marking with a finite number of steps.

The chapter is organized as follows: in Section 2.1 reachability in continuous systems is introduced formally and by means of examples. A preview of the main results is given in that section. Sections 2.2, 2.3 and 2.4 are devoted to the characterization of the sets of reachable markings according to the different concepts: reachability, lim-reachability and δ-reachability respectively. Moreover, it will be seen that it is decidable whether a given marking belongs to any of those three concepts.

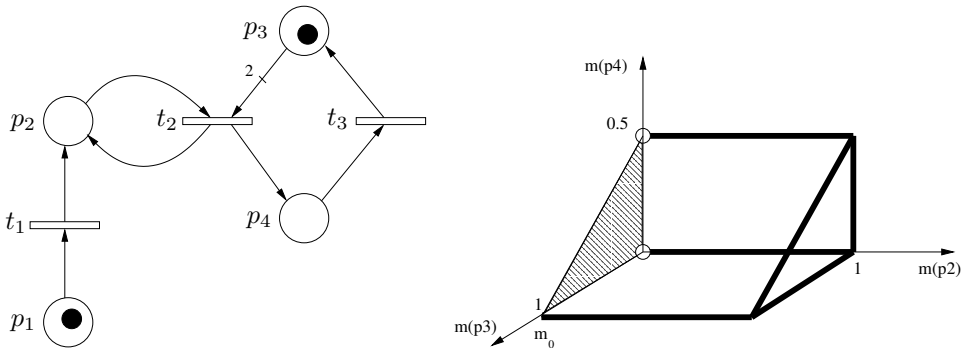
2.1 Definitions and Preview

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

Definition 2.1. $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \{\ \mathbf{m} |$ a finite fireable sequence $\sigma = \alpha_1 t_{a_1} \dots \alpha_k t_{a_k}$ exists such that $\mathbf{m_0} \xrightarrow{\alpha_1 t_{a_1}} \mathbf{m}_1 \xrightarrow{\alpha_2 t_{a_2}} \mathbf{m}_2 \dots \xrightarrow{\alpha_k t_{a_k}} \mathbf{m}_k = \mathbf{m}$ where $t_{a_i} \in T$ and $\alpha_i \in \mathbb{R}^+\}$.

An interesting property of $\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ is that it is a *convex* set (see [RTS99]). That is, if two markings $\mathbf{m}_1$ and $\mathbf{m}_2$ are reachable, then for any $\alpha \in [0, 1]$ $\alpha\mathbf{m}_1 + (1-\alpha)\mathbf{m}_2$ is also a reachable marking. The markings $\mathbf{m}_1 = (0.5,\ 0.5,\ 1,\ 0)$ and $\mathbf{m}_2 = (0.5,\ 0.5,\ 0,\ 0.5)$ are reachable for the system in Figure 2.1(a) by firing respectively t_1 in an amount of 0.5, and t_1 and t_2 in an amount of 0.5. Therefore, since the set of

reachable markings is convex then any marking in the line connecting $\mathbf{m}_1$ and $\mathbf{m}_2$ is also reachable.



(a) (b)

Figure 2.1: (a) Untimed continuous system. (b) Lim-Reachability set.

Let us consider again the system in Figure 2.1(a) with initial marking $\mathbf{m}_0 = (0.5,\ 0.5,\ 0,\ 0.5)$. At this marking either transition t_1 or transition t_3 can be fired. The firing of t_3 in an amount of 0.5 makes the system evolve to marking $(0.5,\ 0.5,\ 0.5,\ 0)$ from which t_2 can be fired in an amount of 0.25 leading to marking $(0.5,\ 0.5,\ 0,\ 0.25)$. Now, the markings of places p_1, p_2 and p_3 are the same as those of the system at $\mathbf{m}_0$, but the marking of p_4 is half of its marking at $\mathbf{m}_0$. The firing of transitions t_2 and t_3 in its maximum enabling degree causes the elimination of half of the marking of p_4. Assume that transitions t_2 and t_3 are further fired. Then, as the number of firings increases the marking of p_4 approaches 0, value that will only be reached in the limit. Notice that the marking reached in the limit $(0.5,\ 0.5,\ 0,\ 0)$ corresponds to the emptying of an initially marked trap ($\Theta = \{p_3, p_4\}, \Theta^\bullet = {}^\bullet\Theta = \{t_2, t_3\}$), fact that does not occur in discrete systems. From the point of view of the analysis of the behaviour of the system, it is interesting to consider this marking as limit-reachable, since in the limit the system may converge to it. Let us define the set of such markings that are reachable with a finite or an infinite firing sequence:

Definition 2.2. [RTS99] Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system. A marking $\mathbf{m} \in (\mathbb{R}^+ \cup \{0\})^{|P|}$ is lim-*reachable*, iff a sequence of reachable markings $\{\mathbf{m}_i\}_{i \geq 1}$ exists such that

$$\mathbf{m_0} \xrightarrow{\sigma_1} \mathbf{m}_1 \xrightarrow{\sigma_2} \mathbf{m}_2 \cdots \mathbf{m}_{i-1} \xrightarrow{\sigma_i} \mathbf{m}_i \cdots$$

and $\lim_{i \to \infty} \mathbf{m}_i = \mathbf{m}$. The lim-reachable set is the set of lim-reachable markings, and will be denoted lim-RS$(\mathcal{N}, \mathbf{m_0})$.

Figure 2.1(b) depicts the lim-reachability set of system in Figure 2.1(a). It is not necessary to represent the marking of place p_1 since $\mathbf{m}_1 = 1 - \mathbf{m}_2$. The set of lim-reachable markings is composed of the points inside the prism, the points in the non shadowed sides, the points in the thick edges and the points in the non circled vertices.

For some systems, the sets RS($\mathcal{N}, \mathbf{m_0}$) and lim-RS($\mathcal{N}, \mathbf{m_0}$) are identical. In that case, with regard to the set of reachable markings, there exists no difference between considering sequences of finite or infinite length. See Figure 2.2 for an example. Only $\mathbf{m}_2$ and $\mathbf{m}_4$ are represented since $\mathbf{m}_1 = 1 - \mathbf{m}_2$ and $\mathbf{m}_3 = 1 - \mathbf{m}_4$. The innner points of the square defined by the vertices (0, 0), (0, 1), (1, 1) and (1, 0), and the thick lines in Figure 2.2(b) are part of the reachability and the lim-reachability set, while the points going from $\mathbf{m_0}$ to (0, 1) (including (0, 1)) do not belong to these sets.

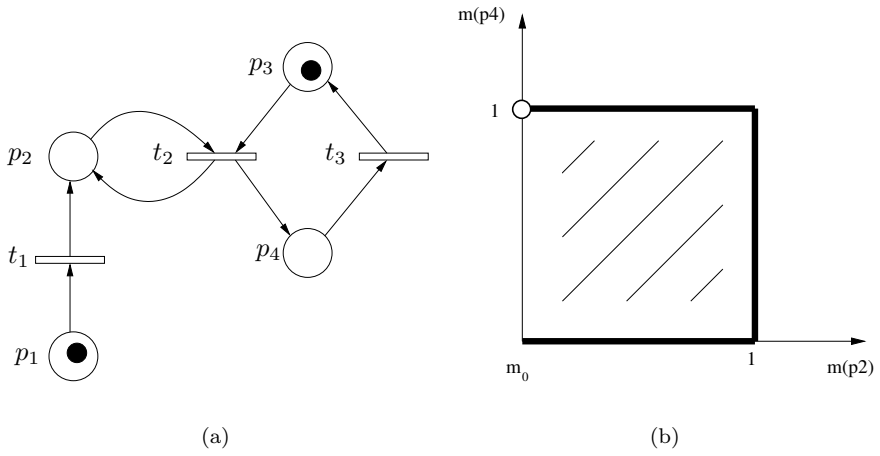


Figure 2.2: (a) Untimed continuous system. (b) Reachability set and Lim-Reachability set coincide.

In general, the set of reachable markings, RS($\mathcal{N}, \mathbf{m_0}$) is a subset of the set of lim-reachable markings, lim-RS($\mathcal{N}, \mathbf{m_0}$). For the system in Figure 2.3(b), neither p_1 nor p_2 can be emptied with a finite firing sequence because every time a transition is fired some marks are put in both places. For that system the set of reachable markings is $(\alpha,\ 2-\alpha),\ 0 < \alpha < 2$. Nevertheless, considering the sequence $\frac{1}{2}t_1, \frac{1}{4}t_1, \frac{1}{8}t_1, \ldots$, in the k-th step, the system reaches the marking $(2^{-k},\ 2-2^{-k})$. When k tends to infinity the marking of the system tends to $(0,\ 2)$. Therefore the infinite firing of t_1 (t_2) will converge to a marking in which p_1 (p_2) is empty. Thus the set of markings reachable in the limit is $(\alpha,\ 2-\alpha),\ 0 \leq \alpha \leq 2$. Notice that the only difference between lim-RS($\mathcal{N}, \mathbf{m_0}$) and RS($\mathcal{N}, \mathbf{m_0}$) is in the markings $(0,\ 2)$ and $(2,\ 0)$. Observe that

even under consistency and conservativeness $\text{RS}(\mathcal{N}, \mathbf{m_0}) \neq \text{lim-RS}(\mathcal{N}, \mathbf{m_0})$.

For the system in Figure 2.3(a), p_1 (p_2) can be emptied with the firing of t_1 (t_2) in an amount of 1. Hence, although the systems in Figure 2.3 have the same incidence matrix, their sets of finitely reachable markings are not the same.

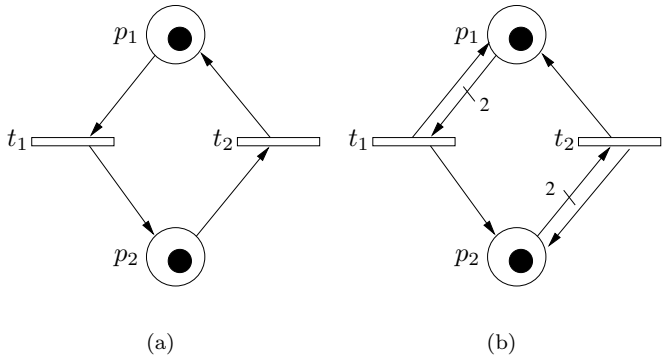


Figure 2.3: Continuous systems that have the same incidence matrix and whose reachability sets do not coincide.

Both $\text{RS}(\mathcal{N}, \mathbf{m_0})$ and $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$ are not in general closed sets. For example in Figure 2.2(b) the points on the segment going from (0, 0) (initial marking) to (0, 1) do neither belong to $\text{RS}(\mathcal{N}, \mathbf{m_0})$ nor to $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$. Nevertheless, any point on the right of this segment does belong to both sets $\text{RS}(\mathcal{N}, \mathbf{m_0})$ and $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$. For a given set A, the closure of A is equal to the points in A plus those points which are infinitely close to points in A, but are not contained in A. In the case of the set depicted in Figure 2.2(b) its closure is equal to the inner and edge points of the square defined by the vertices (0, 0), (0, 1), (1, 1) and (1, 0), that is, it is obtained by adding the segment $[(0,\ 0), (0,\ 1)]$ to $\text{RS}(\mathcal{N}, \mathbf{m_0})$.

Focusing on the sets defined by $\text{RS}(\mathcal{N}, \mathbf{m_0})$ and $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$ and closing them, it will be noticed that the points limiting both sets are exactly the same. This is because if the system can get as close as desired to a given point with an infinite sequence, it can also get as close as desired with a finite sequence and vice versa. Hence, the following property can be stated:

Property 2.3. The closure of $\text{RS}(\mathcal{N}, \mathbf{m_0})$ is equal to the closure of $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$.

Assume that, given a system, $\text{RS}(\mathcal{N}, \mathbf{m_0})$ and $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$ are not identical sets, i.e., $\text{RS}(\mathcal{N}, \mathbf{m_0}) \subsetneq \text{lim-RS}(\mathcal{N}, \mathbf{m_0})$. This means that for every $\mathbf{m}$ in $\text{lim-RS}(\mathcal{N}, \mathbf{m_0}) \setminus \text{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{m}$ is a *border* point of $\text{lim-RS}(\mathcal{N}, \mathbf{m_0})$ and there are markings in $\text{RS}(\mathcal{N}, \mathbf{m_0})$ that are infinitely close to $\mathbf{m}$. Let us make a final consideration on the system of Figure 2.1(a). It has been seen that the initial firing of t_1

enables t_2 and that an infinite sequence consisting on firing t_2 and t_3 will empty p_3 and p_4, reaching marking $(0.5,\ 0.5,\ 0,\ 0)$. In that example t_1 was fired in an amount of 0.5. Nevertheless, p_3 and p_4 can be emptied also if t_1 is fired in an amount α such that $0 < \alpha \leq 1$. For example, if one takes $\alpha = 0.1$, transition t_1 is fired in an amount of 0.1 and then t_2 is fired five times in an amount of 0.1. Now one can fire completely, in an amount of 0.5, transition t_3. Repeating this procedure, in the limit p_3 and p_4 become empty. Thus, it can be said that the marking $(1-\alpha,\ \alpha,\ 0,\ 0)$ is lim-reachable for any α such that $0 < \alpha \leq 1$. Hence, marking $(1,\ 0,\ 0,\ 0)$ is not lim-reachable but the system can get as close as desired to it by taking a small enough α. This marking can then be interpreted as the fact that a little leak of fluid from p_1 to p_2 can cause the emptying of p_3 and p_4. In some situations, it may be useful to consider those markings like $(1,\ 0,\ 0,\ 0)$, that are not reachable, but for which the system can get as close as desired.

Let us consider a norm in order to determine the proximity of two markings. Let $|\mathbf{x}|$ denote the norm of vector $\mathbf{x} = (x_1, \ldots, x_n)$ defined as: $|\mathbf{x}| = |x_1| + \ldots + |x_n|$. A new reachability concept for continuous systems will be introduced: the δ-reachability. The set of δ-reachable markings will be written as δ-RS$(\mathcal{N}, \mathbf{m_0})$ and accounts for those markings to which the system can get as close as desired firing a finite sequence. Formally:

Definition 2.4. δ-RS$(\mathcal{N}, \mathbf{m_0})$ is the closure of RS$(\mathcal{N}, \mathbf{m_0})$: δ-RS$(\mathcal{N}, \mathbf{m_0}) = \{\ \mathbf{m}\ |$ for every $\epsilon > 0$ a marking $\mathbf{m'} \in$ RS$(\mathcal{N}, \mathbf{m_0})$ exists such $|\mathbf{m'} - \mathbf{m}| < \epsilon\}$.

Since the closure of RS$(\mathcal{N}, \mathbf{m_0})$ is equal to the closure of lim-RS$(\mathcal{N}, \mathbf{m_0})$, δ-RS$(\mathcal{N}, \mathbf{m_0})$ is also equal to the set of markings to which the system can get as close as desired firing an infinite sequence. RS$(\mathcal{N}, \mathbf{m_0})$ and lim-RS$(\mathcal{N}, \mathbf{m_0})$ are, therefore, subsets of δ-RS$(\mathcal{N}, \mathbf{m_0})$.

Therefore, till now three different kinds of reachability concepts have been defined:

- Markings that are reachable with a finite firing sequence, RS$(\mathcal{N}, \mathbf{m_0})$.
- Markings to which the system converges, eventually, with an infinitely long sequence, lim-RS$(\mathcal{N}, \mathbf{m_0})$.
- Markings to which the system can get as close as desired with a finite sequence, δ-RS$(\mathcal{N}, \mathbf{m_0})$.

Let us finish this section by defining the linearized reachability set with respect to the state equation:

Definition 2.5. LRS$(\mathcal{N}, \mathbf{m_0}) = \{\mathbf{m} | \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0}$ with $\boldsymbol{\sigma} \in (\mathbb{R}^+ \cup \{0\})^{|T|}\}$.

Notice that given a consistent system (i.e., $\exists\ \mathbf{x} > \mathbf{0} | \mathbf{C} \cdot \mathbf{x} = \mathbf{0}$) it holds: LRS$(\mathcal{N}, \mathbf{m_0}) = \{\mathbf{m} | \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0}$ with $\boldsymbol{\sigma} \in \mathbb{R}^{|T|}\}$. In [RTS99] it was

shown that for consistent systems in which every transition is fireable at least once, the sets LRS($\mathcal{N}, \mathbf{m_0}$) and lim-RS($\mathcal{N}, \mathbf{m_0}$) are the same. This result will be generalized by describing the set of lim-reachable markings of a general system.

By definition LRS($\mathcal{N}, \mathbf{m_0}$) is a closed set. $\mathbf{m}$ is a *border* point of LRS($\mathcal{N}, \mathbf{m_0}$) iff for every $\epsilon > 0$ there exists $\mathbf{m'}$, $|\mathbf{m'} - \mathbf{m}| < \epsilon$ such that $\mathbf{m'} \notin$ LRS($\mathcal{N}, \mathbf{m_0}$).

The *open* set of LRS($\mathcal{N}, \mathbf{m_0}$) is the result of removing every border point from LRS($\mathcal{N}, \mathbf{m_0}$) and will be denoted as]LRS($\mathcal{N}, \mathbf{m_0}$)[.

Notice that given a system $\langle \mathcal{N}, \mathbf{m_0} \rangle$ if there exists $\mathbf{y} \neq \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} = 0$ then every $\mathbf{m} \in$ LRS($\mathcal{N}, \mathbf{m_0}$) is a border point of LRS($\mathcal{N}, \mathbf{m_0}$), and so in this case]LRS($\mathcal{N}, \mathbf{m_0}$)[$= \emptyset$. If such $\mathbf{y}$ exists all the points in LRS($\mathcal{N}, \mathbf{m_0}$) are contained in a hyperplane of smaller dimension than the number of places. In particular, if a system has a P-semiflow, every marking in LRS($\mathcal{N}, \mathbf{m_0}$) is a border point. Those markings having null components are also border points of LRS($\mathcal{N}, \mathbf{m_0}$).

Since all reachable, lim-reachable and δ-reachable markings are solution of the state equation, the following relation is satisfied:

RS($\mathcal{N}, \mathbf{m_0}$) $\subseteq$ lim-RS($\mathcal{N}, \mathbf{m_0}$) $\subseteq$ δ-RS($\mathcal{N}, \mathbf{m_0}$) $\subseteq$ LRS($\mathcal{N}, \mathbf{m_0}$).

Along the chapter this relationship among the different sets will be completed showing that the open linearized set,]LRS($\mathcal{N}, \mathbf{m_0}$)[, is contained in RS($\mathcal{N}, \mathbf{m_0}$) and that δ-RS($\mathcal{N}, \mathbf{m_0}$) = LRS($\mathcal{N}, \mathbf{m_0}$) if every transition is fireable at least once.

2.2 RS($\mathcal{N}, \mathbf{m_0}$)

The goal of this section is first to provide a full characterization of the set of reachable markings (Subsection 2.2.1) and then to show a computation algorithm that decides the reachability of a given target marking (Subsection 2.2.2).

2.2.1 Reachability characterization

Before showing the main result (Theorem 2.12), some intermediate lemmas will be presented in order to ease the final characterization. First, let us introduce an algorithm to compute the sets of transitions fireable from the initial marking, and some interesting results dealing with continuous systems.

Let $FS(\mathcal{N}, \mathbf{m_0})$ be the set of sets of transitions for which there exists a sequence fireable from $\mathbf{m_0}$ that contains those and only those transitions in the set. Formally,

Definition 2.6. $FS(\mathcal{N}, \mathbf{m_0}) = \{ \theta |$ there exists a sequence fireable from $\mathbf{m_0}$, σ, such that $\theta = \|\boldsymbol{\sigma}\| \}$.

Algorithm 2.7 (Computation of the set $FS(\mathcal{N}, \mathbf{m_0})$).

1. Let V be the set of transitions enabled at $\mathbf{m_0}$

2. $FS := \{v | v \subseteq V\}$ % all the subsets of V including the empty set
3. Repeat
  3.1. take $f \in FS$ such that it has not been taken before
  3.2. fire sequentially from $\mathbf{m_0}$ every transition in f without disabling any enabled transition. Let $\mathbf{m}$ be the reached marking.
  3.3. $V := \{t |\ t$ is enabled at $\mathbf{m}$ and $t \notin f\}$
  3.4. $FS := FS \cup \{f \cup v | v \subseteq V\}$
4. until FS does not increase

Notice that step **3.2.** can always be achieved since for any element $f \in FS(\mathcal{N}, \mathbf{m_0})$ there exists a fireable sequence that contains every transition in f. Algorithm 2.7 accounts for all possible subsets of transitions that can become enabled, and so its complexity is exponential on the number of transitions and so is the size of the set $FS(\mathcal{N}, \mathbf{m_0})$. As an example, considering the net in Figure 2.4 with initial marking $\mathbf{m_0} = (1,\ 0,\ 1,\ 1,\ 0)$ the result of Algorithm 2.7 is $FS(\mathcal{N}, \mathbf{m_0}) = \{\ \{\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_2,\ t_3\}, \{t_2,\ t_4\}, \{t_3,\ t_4\}, \{t_2,\ t_3,\ t_4\}, \{t_1,\ t_2\}, \{t_4,\ t_5\}, \{t_1,\ t_2,\ t_3\}, \{t_1,\ t_2,\ t_4\}, \{t_2,\ t_4,\ t_5\}, \{t_1,\ t_2,\ t_4,\ t_5\}, \{t_3,\ t_4,\ t_5\}, \{t_1,\ t_2,\ t_3,\ t_4\}, \{t_2,\ t_3,\ t_4,\ t_5\}, \{t_1,\ t_2,\ t_3,\ t_4,\ t_5\}\}$.

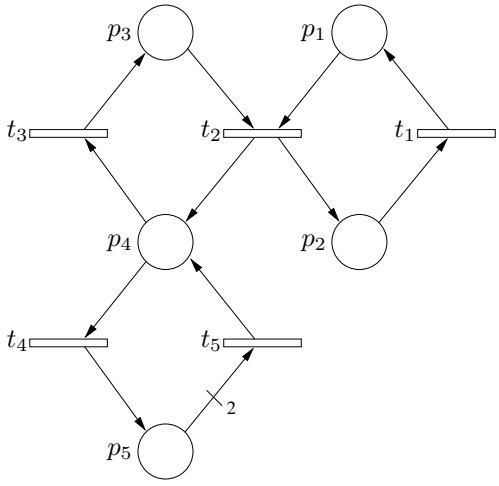


Figure 2.4: Non-consistent continuous system.

Now let us introduce four lemmas that will help to characterize the set of reachable markings. The first one simply states that continuous systems are homothetic with respect to the scaling of $\mathbf{m_0}$.

Lemma 2.8. [RTS99] Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system. If σ is a fireable sequence yielding marking $\mathbf{m}$, then for any $\alpha \geq 0$, $\alpha\sigma$ is fireable at $\alpha\mathbf{m_0}$ yielding marking $\alpha\mathbf{m}$,

where $\alpha\sigma$ represents a sequence that is equal to σ except in the amount of each firing, that is multiplied by α.

Although this section deals with those markings that are reachable with a finite firing sequence, a lemma that has to do with the markings that can be reached in the limit will be presented. Lemma 2.9 establishes that if all the transitions in the support of a given firing vector $\boldsymbol{\sigma}$ are enabled, then $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$ is reachable in the limit, whatever the value of $\boldsymbol{\sigma}$ is. Furthermore, there exists a sequence of reachable markings that are "in the direction" of $\mathbf{m}$.

Lemma 2.9. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system. Let $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$, $\boldsymbol{\sigma} \geq \mathbf{0}$ and $\mathbf{m_0}$ such that for every $t \in \|\boldsymbol{\sigma}\|$ enab$(t, \mathbf{m_0}) > 0$. Then, there exists a succession of reachable markings $\mathbf{m}_1, \mathbf{m}_2, \ldots$ fulfilling $\mathbf{m}_1 = \mathbf{m}_0 + \beta_1 \mathbf{C} \cdot \boldsymbol{\sigma}$, $\mathbf{m}_2 = \mathbf{m}_0 + \beta_2 \mathbf{C} \cdot \boldsymbol{\sigma}, \ldots$ with $0 < \beta_1 < \beta_2 < \ldots$ that converges to $\mathbf{m}$.

Proof. Since at $\mathbf{m_0}$ every transition of $\|\boldsymbol{\sigma}\|$ is enabled, α and $\boldsymbol{\sigma'}$ exist such that $\boldsymbol{\sigma'}$ is fireable from $\mathbf{m_0}$ and $\boldsymbol{\sigma'} = \alpha\boldsymbol{\sigma}$, i.e., a sequence proportional to the vector leading from $\mathbf{m_0}$ to $\mathbf{m}$ can be fired. If $\alpha \geq 1$, it is clear that $\mathbf{m}$ can be reached from $\mathbf{m_0}$. Otherwise, the firing of $\boldsymbol{\sigma'}$ leads to $\mathbf{m_0} + \mathbf{C} \cdot \alpha\boldsymbol{\sigma} = (1-\alpha)\mathbf{m_0} + \alpha\mathbf{m_0} + \mathbf{C} \cdot \alpha\boldsymbol{\sigma} = (1-\alpha)\mathbf{m_0} + \alpha\mathbf{m}$. By Lemma 2.8, if $\boldsymbol{\sigma'}$ was fireable from $\mathbf{m_0}$, then $(1-\alpha)\boldsymbol{\sigma'}$ is fireable from $(1-\alpha)\mathbf{m_0}$. In this way, one obtains

$$\alpha\mathbf{m} + (1-\alpha)\mathbf{m_0} \stackrel{(1-\alpha)\boldsymbol{\sigma'}}{\longrightarrow} \alpha\mathbf{m} + \alpha(1-\alpha)\mathbf{m} + (1-\alpha)^2\mathbf{m_0}$$

Repeating this procedure, in the iteration n the system reaches the marking

$$\alpha\mathbf{m}(1 + (1-\alpha) + (1-\alpha)^2 + \ldots + (1-\alpha)^n) + (1-\alpha)^n\mathbf{m_0}$$

Thus, the marking of the system as n goes to infinity converges to $\mathbf{m}$. □

Based on this result a part of the set of reachable markings can be described.

Lemma 2.10. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system. Let $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$, $\boldsymbol{\sigma} \geq \mathbf{0}$ and for every $t \in \|\boldsymbol{\sigma}\|$ enab$(t, \mathbf{m_0}) > 0$ and enab$(t, \mathbf{m}) > 0$. Then $\mathbf{m} \in RS(\mathcal{N}, \mathbf{m_0})$.

Proof. Every $t \in \|\boldsymbol{\sigma}\|$ is enabled at $\mathbf{m}$. This means that for every $t \in \|\boldsymbol{\sigma}\|$ all its input places are positively marked at $\mathbf{m}$. Then, one can define an $\mathbf{m'}$ such that $\mathbf{m'} = \mathbf{m_0} + \mathbf{C} \cdot (1+\alpha)\boldsymbol{\sigma} \geq 0$ with $\alpha > 0$. According to Lemma 2.9 there is a succession of markings that converges to $\mathbf{m'}$. Since $\mathbf{m}$ is in the line that goes from $\mathbf{m_0}$ to $\mathbf{m'}$ one can stop that sequence at a given step and reach exactly $\mathbf{m}$ in a finite number of firings. □

The following last lemma imposes a necessary and sufficient condition for the fireability of a transition in terms of siphons.

Lemma 2.11. Let $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$. Transition t is not fireable for any successor of $\mathbf{m}$ iff there exists an empty siphon at $\mathbf{m}$ containing a place p such that $p \in {}^\bullet t$.

Proof.
($\Leftarrow$)
If there exists such an empty siphon Θ, no transition in $\Theta^\bullet$ is fireable.
($\Rightarrow$)
Assume t is not fireable for any successor of $\mathbf{m}$. Then there exists a place p such that $p \in {}^\bullet t$ and $\mathbf{m}(p) = 0$. Furthermore, no input transition of p, t', can ever be fired. Hence, for every t' there exists an empty input place p'. Repeating this reasoning a set of empty places Q is obtained. This set Q has the property that all its input transitions (${}^\bullet Q$) are output transitions ($Q^\bullet$). Hence Q is an empty siphon. □

Before going on with the characterization of the set of reachable markings let us make some considerations on the conditions a given marking $\mathbf{m}$ should fulfill in order to be reachable. First of all, it is clear that a necessary condition for $\mathbf{m}$ to be reachable is that it has to be solution of the state equation, that is, there must exist $\boldsymbol{\sigma}$ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$. Furthermore, $\|\boldsymbol{\sigma}\|$ must be in $FS(\mathcal{N}, \mathbf{m_0})$ in order to have a fireable sequence. In Section 2.1 it has been seen that some marked traps can be emptied in a continuous system with the firing of an infinite sequence. If only finite firing sequences are considered, no marked trap can be emptied. Since now the interest lies in finite firing sequences those $\boldsymbol{\sigma}$'s that correspond to a firing count vector that empties (or fills and then empties) a trap have to be explicitly forbidden. As it will be seen, these necessary conditions are also sufficient for a marking to be reachable.

Given a net $\mathcal{N}$ and a firing sequence $\boldsymbol{\sigma}$, let us denote as $\mathcal{N}_{\boldsymbol{\sigma}}$ the net obtained removing from $\mathcal{N}$ the transitions not in the support of $\boldsymbol{\sigma}$ and the resulting isolated places. In other words, $\mathcal{N}_{\boldsymbol{\sigma}}$ is the net composed of the transitions of $\mathcal{N}$ in the support of $\boldsymbol{\sigma}$ and their input and output places. Using the previous lemmas a full characterization of the set of reachable markings is obtained.

Theorem 2.12. A marking $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ iff

1. $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0,\ \boldsymbol{\sigma} \geq 0$
2. $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$
3. there is no empty trap in $\mathcal{N}_{\boldsymbol{\sigma}}$ at $\mathbf{m}$

Proof.
$\subseteq$
Let $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$. Then, there exists $\boldsymbol{\sigma} \geq \mathbf{0}$ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$ and $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$. Furthermore, there cannot be an empty trap in $\mathcal{N}_{\boldsymbol{\sigma}}$ at $\mathbf{m}$ since it would mean that the trap was emptied with a finite firing sequence.
$\supseteq$

Let $\mathbf{m}$ be such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0},\ \boldsymbol{\sigma} \geq \mathbf{0}$, $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$ and there is no empty trap in $\mathcal{N}_{\boldsymbol{\sigma}}$ at $\mathbf{m}$. It will be shown that $\mathbf{m}$ can be reached from $\mathbf{m_0}$ by a finite firing sequence. This will be done in three steps: from $\mathbf{m_0}$ the system will reach a marking $\mathbf{m}'$ at which every transition $t \in \|\boldsymbol{\sigma}\|$ is enabled. From $\mathbf{m}'$ the system will be driven to a marking $\mathbf{m}''$ at which also every transition $t \in \|\boldsymbol{\sigma}\|$ is enabled and is as closed as desired to $\mathbf{m}$. Finally, due to the way $\mathbf{m}''$ is defined it is shown that $\mathbf{m}$ is reachable from $\mathbf{m}''$. Although the order of the sequence of reachable markings is $\mathbf{m_0}$, $\mathbf{m}'$, $\mathbf{m}''$ and $\mathbf{m}$, let us start by defining $\mathbf{m}''$ and showing how it can be reached.

If every $t \in \|\boldsymbol{\sigma}\|$ is enabled at $\mathbf{m}$ then $\mathbf{m}'' = \mathbf{m}$. Otherwise, let us consider the system with marking $\mathbf{m}$ and let us fire backwards and sequentially transitions in $\|\boldsymbol{\sigma}\|$ until a marking ($\mathbf{m}''$) is reached at which every transition in $\|\boldsymbol{\sigma}\|$ is enabled. Notice that this backward firing is equivalent to a forward firing in the reverse net (changing directions of arcs). It will be reasoned that such a firing from $\mathbf{m}$ to $\mathbf{m}''$ is always feasible. Notice that in the reverse net traps have become siphons (structural deadlocks) and the forward firing in the reverse net of transitions in $\|\boldsymbol{\sigma}\|$ never involves the filling of empty siphons of $\mathcal{N}_{\boldsymbol{\sigma}}$ at $\mathbf{m}$. This is because according to the initial condition 3, "there is no empty trap in $\mathcal{N}_{\boldsymbol{\sigma}}$ at $\mathbf{m}$". Therefore, by Lemma 2.11 it can be assured that every transition $t \in \|\boldsymbol{\sigma}\|$ can be fired in the reverse net. Let us denote $\widehat{\boldsymbol{\sigma}}$ the firing count vector such that $\mathbf{m} = \mathbf{m}'' + \mathbf{C} \cdot \widehat{\boldsymbol{\sigma}}$ with $\|\widehat{\boldsymbol{\sigma}}\| = \|\boldsymbol{\sigma}\|$.

Now let us define $\mathbf{m}'$ as a marking reached from $\mathbf{m_0}$ at which every transition $t \in \|\boldsymbol{\sigma}\|$ is enabled, $\mathbf{m}' = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}'$ and $\boldsymbol{\sigma}' \leq \boldsymbol{\sigma}$. This is always possible since $\|\boldsymbol{\sigma}\|$ belongs to $FS(\mathcal{N}, \mathbf{m_0})$ and so one can fire small amounts of the transitions in $\|\boldsymbol{\sigma}\|$ until every transition in $\|\boldsymbol{\sigma}\|$ is enabled. Let us define $\boldsymbol{\sigma}''$ as $\boldsymbol{\sigma}'' = \boldsymbol{\sigma} - \boldsymbol{\sigma}' - \widehat{\boldsymbol{\sigma}}$, then $\mathbf{m}'' = \mathbf{m}' + \mathbf{C} \cdot \boldsymbol{\sigma}''$. Notice that since $\boldsymbol{\sigma}'$ and $\widehat{\boldsymbol{\sigma}}$ can be taken as small as wanted and their supports are contained in the support of $\boldsymbol{\sigma}$, it can always be verified that $\boldsymbol{\sigma}'' \geq 0$ and $\|\boldsymbol{\sigma}\| = \|\boldsymbol{\sigma}''\|$. Moreover, Lemma 2.10 can be directly applied on $\mathbf{m}'$ and $\boldsymbol{\sigma}''$ obtaining that $\mathbf{m}''$ is reachable from $\mathbf{m}'$. And finally, it can be concluded that $\mathbf{m}$ is reachable from $\mathbf{m_0}$. □

Figure 2.5 sketches the trajectory built by the proof of Theorem 2.12 to reach $\mathbf{m}$.

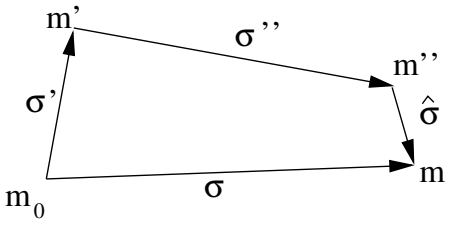


Figure 2.5: Trajectory to reach $\mathbf{m}$ with a finite firing sequence.

As an example, let us take the system in Figure 2.6. The marking $\mathbf{m}$ =

(0, 0, 0, 0, 1) is solution of the state equation and can be obtained with vectors: $\boldsymbol{\sigma}_1 = (1, 0, 1, 1, 0, 0)$ and $\boldsymbol{\sigma}_2 = (0, 1, 0, 0, 1, 0)$. Obviously, $\boldsymbol{\sigma}_2$ fulfills the conditions of Theorem 2.12, and so it can be concluded that $\mathbf{m}$ can be reached. However, if one considers the system that results of removing transitions t_2, t_5 and the place p_4, then the only possibility to reach $\mathbf{m}$ is with $\boldsymbol{\sigma}_1$ or with $\boldsymbol{\sigma}_1 + \mathbf{x}$ where $\mathbf{x}$ is a T-semiflow. Notice that the nets $\mathcal{N}_{\boldsymbol{\sigma}_1}$ and $\mathcal{N}_{\boldsymbol{\sigma}_1+\mathbf{x}}$ have an empty trap at $\mathbf{m}$ composed of $\{p_2, p_3\}$. Hence, the third condition of Theorem 2.12 is violated and $\mathbf{m}$ cannot be reached with a finite sequence.

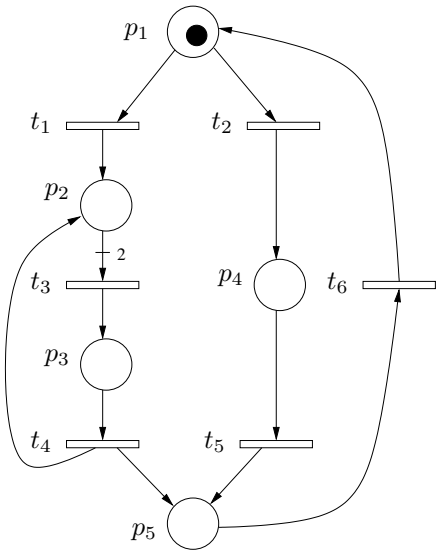


Figure 2.6: Marking (0, 0, 0, 0, 1) can be reached with a finite and with an infinite firing sequence.

2.2.2 Deciding reachability

Based on Theorem 2.12, an algorithm that decides whether a given marking $\mathbf{m}$ is reachable or not is introduced. A necessary condition for $\mathbf{m}$ to be reached is that there must exist a $\boldsymbol{\sigma} \geq 0$ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$. Given a marking $\mathbf{m}$ the number of $\boldsymbol{\sigma} \geq 0$ fulfilling the state equation can be infinite. However, as stated in Theorem 2.12, it is only interesting to consider those $\boldsymbol{\sigma}$'s such that $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$. Furthermore, it is not necessary to consider two different $\boldsymbol{\sigma}$'s that are solution of the state equation and have the same support, since clearly one of those $\boldsymbol{\sigma}$'s fulfills the condition on the traps of Theorem 2.12 iff the other one also fulfills it, and the support of one belongs to $FS(\mathcal{N}, \mathbf{m_0})$ iff the other one also belongs to it. This reasoning reduces the number

of $\boldsymbol{\sigma}$'s to be considered to a finite number.

Now let us take into account a set Σ of $\boldsymbol{\sigma}$'s that are solution of the state equation, have different supports and the support of all of them is in $FS(\mathcal{N}, \mathbf{m_0})$. To decide reachability it is only necessary to consider those $\boldsymbol{\sigma}$'s with minimal support. This is because if there is a non minimal $\boldsymbol{\sigma} \in \Sigma$ fulfilling the condition on the traps of Theorem 2.12, then its support contains the support of a $\boldsymbol{\sigma'} \in \Sigma$ that also fulfills this condition.

Summing up, to decide reachability it is only necessary to consider the $\boldsymbol{\sigma}$'s in a set $\Sigma = \{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k\}$ that fulfills the following conditions:

1. $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}_i \geq 0$ and $\|\boldsymbol{\sigma}_i\| \in FS(\mathcal{N}, \mathbf{m_0})$
2. $\|\boldsymbol{\sigma}_i\|$ is minimal, i.e., for every $j \neq i$ $\|\boldsymbol{\sigma}_j\| \not\subseteq \|\boldsymbol{\sigma}_i\|$
3. for every $\boldsymbol{\gamma}$ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\gamma} \geq 0$ and $\|\boldsymbol{\gamma}\| \in FS(\mathcal{N}, \mathbf{m_0})$ there exists $i \in \{1, \ldots, k\}$ such that $\|\boldsymbol{\sigma}_i\| \subseteq \|\boldsymbol{\gamma}\|$

The third condition guarantees that every $\boldsymbol{\sigma}$ that verifies the first two conditions is included in Σ.

The following algorithm takes as inputs a continuous system, a target marking $\mathbf{m}$, and a set Σ verifying the above conditions for the target marking. The output of the algorithm is the boolean variable *answer* that takes the value *YES* iff $\mathbf{m}$ is reachable. The general idea of the algorithm is first checking whether all traps are marked at $\mathbf{m}$, step 2, and whether there is an empty trap at $\mathbf{m}$ that was marked at $\mathbf{m_0}$, step 3. In these both cases a quick answer to reachability can be given. Otherwise, it is required to iterate on the elements of Σ.

Algorithm 2.13.

INPUT: $\langle \mathcal{N}, \mathbf{m_0} \rangle$, $\mathbf{m}$, $\Sigma = \{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k\}$

OUTPUT: answer

1. If $\Sigma = \emptyset$ then answer:=NO; exit; end if

2. If there is no empty trap in $\mathcal{N}$ at $\mathbf{m}$ then answer:=YES; exit; end if

3. If there is an empty trap in $\mathcal{N}$ at $\mathbf{m}$ that was not empty at $\mathbf{m_0}$ then answer:=NO; exit; end if

4. i:=0

5. loop

  5.1. i:=i+1

  5.2. If there is no empty trap in $\mathcal{N}_{\boldsymbol{\sigma}_i}$ at $\mathbf{m}$ then answer:=YES; exit; end if

6. until i=k

7. answer:=NO

In [ECS93] a method to compute traps based on the solution of a system of linear equations was proposed. According to this method the support of a solution of that system represents the places of a trap. In steps 2 and 5.2 of Algorithm 2.13, the interest lies only on empty traps at $\mathbf{m}$, therefore only the subnet composed of empty places at $\mathbf{m}$ has to be considered. In step 3, the focus is on the empty traps at $\mathbf{m}$ that were marked at $\mathbf{m_0}$. The only thing that has to be included in the system of inequalities proposed in [ECS93] is forcing that a solution of the system must have at least one non null component corresponding to a non empty place at $\mathbf{m_0}$.

2.3 lim-RS$(\mathcal{N}, \mathbf{m_0})$

As it has been shown, some traps (for example the one composed of p_3 and p_4 in the system of Figure 2.1(a)) can be emptied with an infinite firing sequence. Hence when facing the problem of describing the set of lim-reachable markings, it is not necessary to exclude those markings that are result of the state equation and that have empty traps that were previously filled. In this way, the characterization of the lim-RS$(\mathcal{N}, \mathbf{m_0})$ is easier and it is only necessary to care about the fireability of the firing count vector $\boldsymbol{\sigma}$ (conditions 1. and 2. of Theorem 2.12).

Theorem 2.14. A marking $\mathbf{m} \in$ lim-RS$(\mathcal{N}, \mathbf{m_0})$ iff

1. $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0,\ \boldsymbol{\sigma} \geq 0$
2. $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$

Proof.
$\subseteq$
Let $\mathbf{m} \in$ lim-RS$(\mathcal{N}, \mathbf{m_0})$. Since $\mathbf{m}$ is reached by a finite or infinite firing sequence there must exist a firing count vector, $\boldsymbol{\sigma} \geq \mathbf{0}$, corresponding to this sequence such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$. If the sequence was fireable then $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$.
$\supseteq$
Let $\mathbf{m}$ be such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$, $\boldsymbol{\sigma} \geq 0$ and $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$. From $\mathbf{m_0}$ it is possible to fire sequentially a subset of transitions in $\|\boldsymbol{\sigma}\|$, since it belongs to $FS(\mathcal{N}, \mathbf{m_0})$, reaching marking $\mathbf{m'} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma'}$ at which every transition in $\|\boldsymbol{\sigma}\|$ is enabled. Since $\boldsymbol{\sigma'}$ can be taken arbitrarily small, it can always fulfill $\boldsymbol{\sigma} - \boldsymbol{\sigma'} \geq 0$. Lemma 2.9 can be applied on the system $(\mathcal{N}, \mathbf{m'})$ and therefore marking $\mathbf{m}$ can be reached in the limit. □

According to Theorem 2.14 checking whether a given marking is reachable in the limit is a *decidable* problem. For the system in Figure 2.6 without transitions t_2, t_5 and place p_4 it can can be assured that the marking $\mathbf{m} = (0,\ 0,\ 0,\ 0,\ 1)$ is lim-reachable (but not reachable) since it is solution of the state equation with $\boldsymbol{\sigma} = (1,\ 0,\ 1,\ 1,\ 0,\ 0)$ and $\|\boldsymbol{\sigma}\| \in FS(\mathcal{N}, \mathbf{m_0})$.

If the system fulfills some initial conditions, then the set lim-RS($\mathcal{N}$,$\mathbf{m_0}$) can be described without the use of $FS(\mathcal{N}, \mathbf{m_0})$. For example, for a system, $\langle \mathcal{N}, \mathbf{m_0} \rangle$, in which every transition is enabled at $\mathbf{m_0}$, it holds $FS(\mathcal{N}, \mathbf{m_0}) = \{q | q \subseteq T\}$ and therefore every $\boldsymbol{\sigma} \geq \mathbf{0}$ belongs to $FS(\mathcal{N}, \mathbf{m_0})$. Furthermore, this condition can be checked in *polynomial* time.

Corollary 2.15. If for every transition t enab$(t, \mathbf{m_0}) > 0$ then lim-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.

Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a consistent system in which every transition is fireable at least once, i.e., for every transition t there exists $\mathbf{m'} \in$ RS$(\mathcal{N}, \mathbf{m_0})$ such that enab$(t, \mathbf{m'}) > 0$. Clearly $T \in FS(\mathcal{N}, \mathbf{m_0})$. Since the system is consistent it has a T-semiflow $\mathbf{x} > \mathbf{0}$ that can be added to a given $\boldsymbol{\sigma}$, $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0}$, fulfilling $\boldsymbol{\sigma} + \mathbf{x} > \mathbf{0}$. It is obvious that $\mathbf{C} \cdot \boldsymbol{\sigma} = \mathbf{C} \cdot (\boldsymbol{\sigma} + \mathbf{x})$ and that $\|\boldsymbol{\sigma} + \mathbf{x}\| = T$. Therefore, $\mathbf{m}$ is lim-reachable.

Corollary 2.16 ([RTS99]). If $(\mathcal{N}, \mathbf{m_0})$ is consistent and every transition is fireable at least once, then lim-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.

2.4 δ-RS$(\mathcal{N}, \mathbf{m_0})$

Let us now assume that given a system, $\langle \mathcal{N}, \mathbf{m_0} \rangle$, every transition is fireable at least once. That is for every transition t there exists $\mathbf{m} \in$ RS$(\mathcal{N}, \mathbf{m_0})$ such that enab$(t, \mathbf{m}) > 0$. The existence of transitions that do not fulfill this condition can be easily detected (see [RTS99]): It is sufficient to iterate on the enabled transitions firing them in half its enabling degree until no more transitions become enabled. Those transitions that are not enabled after the iteration can never be fired. Notice that this assumption does not imply a loss of generality in the following results, since if a transition can never be enabled it can be removed without affecting any possible evolution of the system or changing the set of reachable markings.

In this section the set of markings to which the system can get as close as desired is described. For example, in Figure 2.4 with $\mathbf{m_0} = (1,\ 0,\ 0,\ 0,\ 1)$, $\mathbf{m} = (0,\ 1,\ 0,\ 0,\ 1)$ does not belong neither to RS($\mathcal{N}$,$\mathbf{m_0}$) nor to lim-RS($\mathcal{N}$,$\mathbf{m_0}$), however $\mathbf{m} = (0,\ 1,\ 0,\ \alpha,\ 1 - 2 \cdot \alpha)$ belongs to RS($\mathcal{N}$,$\mathbf{m_0}$) (hence also to lim-RS($\mathcal{N}$,$\mathbf{m_0}$)) for every α fulfilling $0 < \alpha \leq 0.5$.

For this set of markings, that will be called δ-reachable, there are no spurious solutions of the state equation.

Theorem 2.17. If every transition is fireable at least once from the initial marking, then a marking $\mathbf{m} \in \delta$-RS$(\mathcal{N}, \mathbf{m_0})$ iff

$$1.\ \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0,\ \boldsymbol{\sigma} \geq 0$$

i.e., δ-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.

Proof.
$\subseteq$
δ-RS$(\mathcal{N}, \mathbf{m_0}) \subseteq$ LRS$(\mathcal{N}, \mathbf{m_0})$ since LRS$(\mathcal{N}, \mathbf{m_0})$ is a closed set that includes the RS$(\mathcal{N}, \mathbf{m_0})$.
$\supseteq$
Let $\mathbf{m}$ be a solution of the state equation, i.e., $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0$. Since every transition is fireable at least once, let us consider a sequence, $\boldsymbol{\sigma}'$, that reaches a marking, $\mathbf{m}'$, at which every transition in the support of $\boldsymbol{\sigma}$ is enabled. Let us consider the real quantity α determined by $\alpha = \min\{1, \max\{\beta | \mathbf{m}' + \mathbf{C} \cdot \beta \cdot \boldsymbol{\sigma} \geq 0\}\}$. Then, according to Theorem 2.14, the marking $\mathbf{m}'' = \mathbf{m}' + \mathbf{C} \cdot \alpha \cdot \boldsymbol{\sigma}$ is reachable in the limit from $\mathbf{m}'$. And clearly, it is also reachable in the limit from $\mathbf{m_0}$ ($\mathbf{m}'' \in$ lim-RS$(\mathcal{N}, \mathbf{m_0})$). Notice that if $|\boldsymbol{\sigma}'|$ tends to zero, then the value of α goes to one and $\mathbf{m}''$ approaches $\mathbf{m}$. Thus, by firing a finite sequence the system can get as close to $\mathbf{m}$ as desired. □

Establishing a bridge to discrete systems, it can be said that if the system is highly populated and it is not necessary to exactly determine the marking at places, then the system can evolve to any marking that is solution of the state equation.

Summarizing on reachability, the following relationship among the different sets of reachable markings can be stated. It asserts that the only differences among the described sets of reachable markings are in the border points of the space defined by the state equation.

Corollary 2.18. If every transition is fireable then:

1. $]$LRS$(\mathcal{N}, \mathbf{m_0})[\subseteq$ RS$(\mathcal{N}, \mathbf{m_0}) \subseteq$ lim-RS$(\mathcal{N}, \mathbf{m_0}) \subseteq \delta$-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.
2. Under consistency of $\mathcal{N}$: lim-RS$(\mathcal{N}, \mathbf{m_0}) = \delta$-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.

Proof. 1. is a direct consequence of the fact that δ-RS$(\mathcal{N}, \mathbf{m_0})$ is the closure of RS$(\mathcal{N}, \mathbf{m_0})$ and lim-RS$(\mathcal{N}, \mathbf{m_0})$, and δ-RS$(\mathcal{N}, \mathbf{m_0}) =$ LRS$(\mathcal{N}, \mathbf{m_0})$.

2. is immediate from Corollary 2.16 and Theorem 2.17. □

2.5 Conclusions

In continuous nets the concept of "reachable marking" can be interpreted in three different ways:

a) *reachability*, a marking can be reached with a finite firing sequence.

b) lim-*reachability*, a marking can be reached with a finite or with an infinite firing sequence.

c) δ-*reachability*, the system can get as close as desired to a marking with a finite firing sequence.

Each of the three concepts has its own reachability set. These reachability sets can be fully characterized using, among other elements, the state equation. Moreover, it is decidable whether a marking is reachable according to each concept. Furthermore, there is an inclusion relationship among the sets of markings associated to each concept: $a \subseteq b \subseteq c$. The only differences among these sets are in the border points of the sets (i.e., the convex hull).

As the level of "exigency" regarding reachability decreases (a is the "strongest" and c the "weakest") the characterization of the reachability set becomes progressively easier. In particular, if every transition is fireable at least once, a very weak condition because unfireable transitions can be simply removed, the set of the markings in c is equal to the solutions of the state equation. In other words, for this last case there exist no spurious solutions of the state equation.

Chapter 3

Liveness in Untimed Systems

Summary

Liveness is a very desirable system property that ensures that any transition can be eventually fired from any reachable marking. Unfortunately, liveness may not be preserved when the discrete model is *continuized*. Therefore, a detailed study of liveness in continuous Petri nets is required. In general, liveness analysis of untimed continuous systems is a difficult problem. For mono-T-semiflow systems, the equivalence between liveness and deadlock-freeness allows a more satisfactory treatment. For mutual lim-reachability and δ-reachability (concepts introduced in Chapter 2) among markings, i.e., reversibility, a necessary and sufficient condition is provided in terms of liveness.

Introduction

The continuization of some Petri net systems may yield a non appropriate continuous model: some qualitative properties as liveness may not be preserved [RTS99]. Hence, it becomes interesting to analyze continuous Petri nets in order to establish when a discrete Petri net allows a "reasonable continuization", i.e., when it preserves the properties to be analyzed.

This chapter concentrates on the analysis of liveness on the subclass of *mono-T-semiflow* (MTS) *Petri nets*[JRS02, JRS]. Untimed continuous Petri nets will be considered throughout this chapter. One important property of discrete (and continuous) MTS systems is that deadlock-freeness is equivalent to liveness [CCS91], because all the infinite behaviors are "essentially conformed" by infinite repetition of sequences having (multiples of) the minimal T-semiflow as the firing count vector. Even more, for MTS systems, deadlock-freeness of the untimed model leads, for any arbitrary transition-time semantics (deterministic, exponential, coxian...), to non null throughput. Thus there exists an interesting one-way bridge from logical or qualitative properties to performance or quantitative properties. This fact will be exploited in Chapter 4 where a liveness condition for untimed systems is inferred from a liveness condition for timed systems.

The chapter is structured as follows: In Section 3.1 it is shown that the property of liveness can be lost by the continuized model. In Section 3.2, the concept of lim-liveness is introduced as a natural extension of liveness for discrete systems. In that section, some necessary conditions for lim-liveness in untimed systems are obtained as well. Section 3.3 establishes a tight relationship between liveness and reversibility.

3.1 No liveness preservation in untimed systems

Due to the integrality relaxation in the firing of transitions, the set of potentially reachable markings in an untimed continuous system is much larger than the set of reachable markings if the same system is interpreted as discrete. This fact may have serious consequences in the property of liveness for continuous net systems.

Figure 3.1 shows examples of untimed MTS systems in which liveness of the discrete model is neither necessary nor sufficient for liveness of the relaxed continuous approximation [RTS99]. The system in Figure 3.1(a) deadlocks as discrete after the firing of transition t_1. However, it never gets completely blocked as continuous unless an infinitely long sequence is considered. On the other hand, the system in Figure 3.1(b) is live as discrete but gets blocked as continuous if transition t_2 is fired in an amount of 0.5. This *non-continuizability* of discrete net systems, that may be surprising at first glance, can be easily accepted if one thinks, for example, on the existence of non-linearizable differential equations systems (for example, due to the

existence of a chaotic behavior).

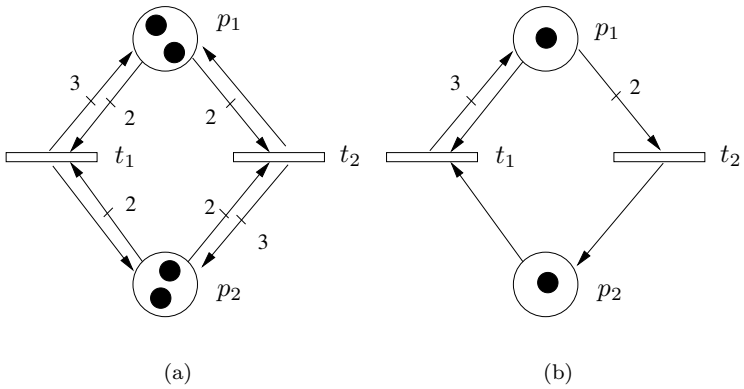


Figure 3.1: Two MTS systems which behave in very different ways if seen as discrete or as continuous.

3.2 lim-liveness in untimed systems

3.2.1 Deadlock-freeness and lim-liveness definition

As shown in Chapter 2, in continuous systems, it may happen that a marking cannot be reached with a finite firing sequence, but there exists an infinite firing sequence that converges to that marking. As an example of this phenomenon, let us consider the system in Figure 3.1(a). The marking $\mathbf{m}[p_1] = 0$, $\mathbf{m}[p_2] = 4$ cannot be reached with a finite firing sequence. However, if transition t_2 could be fired indefinitely in an amount equal to its enabling degree, the marking of p_1 and p_2 would converge to 0 and 4 respectively. According to Chapter 2, $\mathbf{m} = (0\ 4)$ is lim-reachable. At that marking, no transition can be further fired, i.e., the system deadlocks. In practice, a well designed system should have no deadlock markings that are reached either with an infinite firing sequence or with a finite firing sequence. Such deadlock markings represent a non desirable system behaviour towards which the net system may converge. This section is devoted to the study of lim-liveness, i.e., liveness according to the lim-reachability concept.

Definition 3.1. [RTS99] Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous PN system.

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

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

The study of lim-liveness is particularly interesting because liveness of a discrete system is strongly related to lim-liveness of its fluidified version: In [Rec98] it was shown that if $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is a bounded lim-live continuous system then $\mathcal{N}$ is structurally live as a discrete net.

3.2.2 Conditions for lim-liveness for MTS nets

In MTS systems any subset of transitions, $T' \subsetneq T$, can be disabled just by firing (indefinitely) every transition in T'.

Lemma 3.2. Let $\mathcal{N}$ be a MTS net. For every $\mathbf{m_0}$ and every $T' \subsetneq T$ a marking $\mathbf{m} \in$ lim-RS$(\mathcal{N}, \mathbf{m_0})$ exists such that for all $t \in T'$ $enab(t, \mathbf{m}) = 0$. Moreover, this marking can be reached firing only transitions in T'.

Proof. Let $f_0 = \max_{t \in T'}\{enab(t, \mathbf{m_0})\}$. Let us fire sequentially every transition in T' in an amount equal to its enabling degree (maximum firing amount). This action yields a new marking $\mathbf{m}_1$. Let us recompute $f_1 = \max_{t \in T'}\{enab(t, \mathbf{m}_1)\}$. If after repeating these steps a finite number of times it is obtained that every $t \in T'$ is disabled, then the result is proved. Otherwise the procedure can be repeated without limit. Let us suppose that $lim(f_n)_{n \to \infty} = k > 0$. That would mean that there exists a repetitive sequence different from the T-semiflow. Contradiction. □

Notice that disabling a subset of transitions is not equivalent to killing them, since they could be enabled if other transitions not contained in the subset of disabled transitions were fired.

Lemma 3.2 leads to the equivalence between lim-deadlock-freeness and lim-liveness for continuous systems, something well-known for the discrete case [CCS91].

Property 3.3. A continuous MTS system is lim-live iff it is lim-deadlock-free.

Proof. Assume $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is not lim-live. There is a transition t that cannot be fired from any successor of a certain reachable marking. The application of Lemma 3.2 on transitions $T - \{t\}$ leads to a deadlock. □

Suppose that in a given system, $\langle \mathcal{N}, \mathbf{m_0} \rangle$, there is a transition, t, such that for any reachable marking t is never the only enabled transition. This means that if the rest of transitions, $T - \{t\}$, are disabled at a given marking $\mathbf{m}$, then t is also disabled at $\mathbf{m}$. Since every transition of the set $T - \{t\}$ can be disabled in the limit (Lemma 3.2), it can be inferred that $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is not lim-live.

Theorem 3.4. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a lim-live MTS system. For every transition t, $\exists\, \mathbf{m} \in$ lim-RS$(\mathcal{N}, \mathbf{m_0})$ such that t is the only enabled transition at $\mathbf{m}$.

Theorem 3.4 establishes a necessary lim-liveness condition that is illustrated in Figure 3.2. In that system, for every reachable marking in which t_2 is enabled either t_3 or t_4 is enabled. Hence, t_2 is never unavoidably "forced" to fire. Firing several times t_3 and t_4 a deadlock is reached.

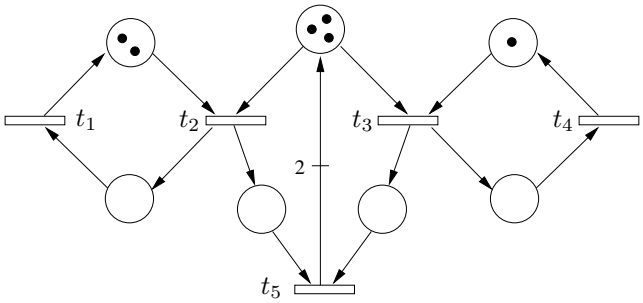


Figure 3.2: A non lim-live system according to Theorem 3.4.

Although the condition given by Theorem 3.4 is in general not easy to check, a simple structural condition (i.e., applicable independently of the initial marking) necessary for liveness can be extracted at net level.

Corollary 3.5. Let $\mathcal{N}$ be a MTS net. If $\mathcal{N}$ is str. lim-live then for every $t \neq t'$, ${}^\bullet t \not\subseteq {}^\bullet t'$ (i.e., preconditions of transitions are non comparable)

Proof. If there exist $t \neq t'$ such that ${}^\bullet t \subseteq {}^\bullet t'$, for every marking in which t' is enabled, t is also enabled. Thus, Theorem 3.4 can be directly applied and non lim-liveness for an arbitrary initial marking is deduced. □

Hence, topological conflicts in which the set of input places of one transition is contained in the set of input places of other transition must be forbidden if the system is wanted to be lim-live as untimed. For example in the system in Figure 3.3, for any reachable marking if t_2 is enabled then t_1 is also enabled. Firing t_2 and t_3 in a long enough sequence a deadlock is reached for any initial marking. Remarkably this system is live if seen as discrete.

In other words, Corollary 3.5 detects a kind of "structural contradiction" in the MTS net: on the one hand all the transitions are included in the only repetitive sequence (the T-semiflow), and on the other hand there exist $t \neq t'$ such that ${}^\bullet t \subseteq {}^\bullet t'$, thus, the net gives the possibility of never firing transition t'. The result of this contradiction entails a deadlock.

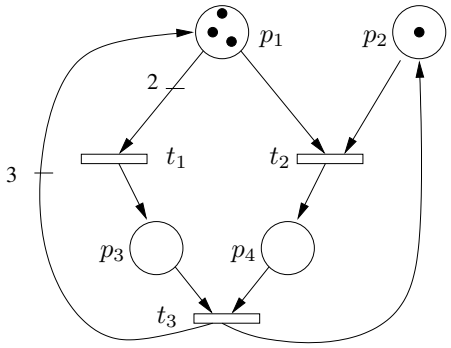


Figure 3.3: A system for which Corollary 3.5 detects non lim-liveness.

In Chapter 4 liveness conditions for timed systems are studied. Some of the results obtained in that chapter can be applied to untimed systems. In particular, Section 4.6 shows how Corollary 3.5 can be improved by making use of a necessary liveness condition for timed systems.

3.3 Reversibility and δ-liveness

Reversibility is a basic property that has to do with mutual reachability among all markings of the system, or equivalently with the ability to reach the initial marking from any reachable one. In discrete systems *if every transition is fireable at least once, then reversibility implies liveness and consistency*: if a system is reversible, it can always get back to the initial marking, therefore it is live because from the initial marking every transition is fireable at least once. Moreover, if a system can always return to the initial marking after every transition has fired, it means that a T-semiflow covering every transition has been fired, that is, the system is consistent.

However, liveness and consistency are not sufficient conditions for reversibility in discrete systems. For example, the system in Figure 3.4 is consistent and live as discrete, however once t_1 has fired it is impossible to get back to the initial marking. Thus the system is not reversible as discrete.

In continuous systems, assuming that every transition is fireable at least once, it can be observed that reversibility also implies consistency and liveness. As in discrete systems, if reachability with finite sequences is considered, liveness and consistency are not sufficient conditions for reversibility. The system in Figure 2.6 is consistent and live as continuous considering finite firing sequences. If transition t_1 is fired in any amount, the trap $\{p_2, p_3, p_4\}$ becomes marked, and cannot be emptied with a finite firing sequence. Hence, once t_1 has fired it is not possible to go back to the initial marking, and therefore it can be said that the system is not reversible. Nevertheless,

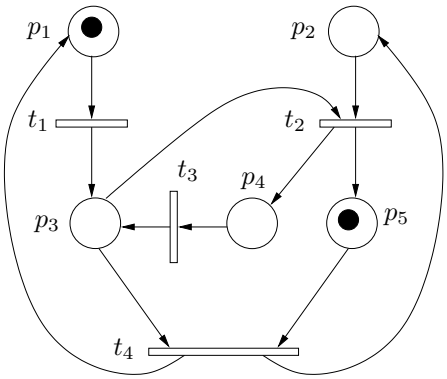


Figure 3.4: Non reversible system as discrete or continuous with finite number of firings, but lim-reversible and δ-reversible.

as it will be seen, the system is reversible if lim-reachability and δ-reachability are considered.

Let us define δ-liveness and lim-(δ-)reversibility as the natural extensions of the classical definitions for the concepts of lim-reachability and δ-reachability respectively:

Definition 3.6. $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is δ-live iff for every $\mathbf{m} \in \delta\text{-RS}(\mathcal{N}, \mathbf{m_0})$ and for every $t \in T$ there exists $\mathbf{m'} \in \text{RS}(\mathcal{N}, \mathbf{m})$ such that $\text{enab}(t, \mathbf{m'}) > 0$.

Definition 3.7.$\langle \mathcal{N}, \mathbf{m_0} \rangle$ is lim-(δ-)reversible iff for every $\mathbf{m} \in$ lim-(δ-)$\text{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{m_0} \in$ lim-(δ-)$\text{RS}(\mathcal{N}, \mathbf{m})$.

The following theorem states that under lim-reachability and δ-reachability, if every transition can be fired at least once, consistency and lim-(δ-)liveness are not only necessary conditions for lim-(δ-)reversibility but also sufficient.

Theorem 3.8. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be such that every transition is fireable at least once. $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is consistent and lim-(δ-)live iff $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is lim-(δ-)reversible.

Proof.
(⇒)
Since the system is consistent and every transition is fireable at least once, it holds by Corollary 2.18 that $\text{lim-RS}(\mathcal{N}, \mathbf{m_0}) = \delta\text{-RS}(\mathcal{N}, \mathbf{m_0}) = \text{LRS}(\mathcal{N}, \mathbf{m_0})$. Let us consider the lim-(δ-)reachable marking $\mathbf{m}$, $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$. It will be seen that $\mathbf{m_0}$ is lim-(δ-)reachable from $\mathbf{m}$. Since the system is lim-(δ-)live, every transition is fireable from $\mathbf{m}$, and therefore a strictly positive marking, $\mathbf{m'} > \mathbf{0}$, can be reached, $\mathbf{m'} = \mathbf{m_0} + \mathbf{C} \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma'})$. The net is consistent, hence a T-semiflow, $\mathbf{x} > \mathbf{0}$, exists such that $\mathbf{x} - \boldsymbol{\sigma} - \boldsymbol{\sigma'} \geq \mathbf{0}$. By Corollary 2.15, $\mathbf{m_0} = \mathbf{m'} + \mathbf{C} \cdot (\mathbf{x} - \boldsymbol{\sigma} - \boldsymbol{\sigma'})$ is lim-(δ)reachable from $\mathbf{m'}$.

($\Leftarrow$)
If the system is reversible and every transition can be fired at least once, then it clearly cannot lim-(δ-)reach a marking in which one transition is not fireable any more. It would mean that it cannot get back to the initial marking. Moreover, if after the firing of every transition the system always can return to the initial marking, it means that it is consistent. □

For example, the system in Figure 3.4 is consistent and lim-(δ-)live, therefore according to Theorem 3.8 it is lim-(δ-)reversible. If from the initial marking t_1 is fired in an amount of 1, the marking (0, 0, 1, 0, 1) is reached. Applying the infinite firing sequence $\frac{1}{2}, t_4\frac{1}{2}t_2, \frac{1}{2}t_3, \frac{1}{4}, t_4\frac{1}{4}t_2, \frac{1}{4}t_3, \ldots$ from (0, 0, 1, 0, 1) the system converge to the initial marking.

From Theorem 3.8, the following Corollary is immediate:

Corollary 3.9. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be lim-(δ-)live. $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is lim-(δ-)reversible iff $\mathcal{N}$ is consistent.

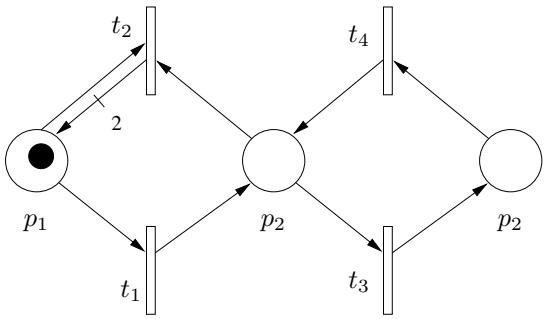


Figure 3.5: A lim-(δ-)deadlock-free and not lim-(δ-)live system.

Notice that for non MTS systems, the lim-(δ-)liveness condition in Theorem 3.8 and Corollary 3.9 cannot be relaxed to lim-(δ-)deadlock-freeness, where lim-(δ-) deadlock-freeness means that the system cannot lim-(δ-)reach a marking in which no transition is fireable. In other words, as in discrete systems, for non MTS systems lim-(δ-)deadlock-freeness does not imply lim-(δ-)liveness, even under consistency and conservativeness. For example, the consistent and conservative system in Figure 3.5 is lim-(δ-)deadlock-free but not lim-(δ-)live: transitions t_3 and t_4 are potentially fireable from any lim-(δ-)reachable marking, but once t_1 is fired in an amount of 1, neither t_1 nor t_2 will ever be fireable.

3.4 Conclusions

Continuous Petri nets can be used to overcome the state explosion problem of highly populated discrete systems. Unfortunately, not every discrete system can be suitably continuized if the property of liveness is wanted to be preserved by the continuized system. In this chapter liveness has been studied in the framework of untimed continuous MTS systems, i.e., systems that are consistent, conservative and have only one T-semiflow.

Since MTS systems have only one repetitive sequence, lim-deadlock-freeness becomes equivalent to lim-liveness. From an untimed point of view, that is, without time interpretation, necessary structural lim-liveness conditions have been obtained. For MTS systems these conditions are more accurate than the one established by the rank Theorem [RTS98]. It is remarkable that those conditions (Theorem 3.4, Corollary 3.5) are stated on topological features of the net, hence disregarding the arc weights. This is a logical consequence of the continuous firing of transitions: The arc weights have an influence on the amount in which one transition is enabled (enabling degree) but deciding whether a transition is enabled or not does not depend on the value of its input arc weights. The last section of the chapter provides a necessary and sufficient condition for reversibility with respect to lim-reachability and δ-reachability.

Chapter 4

Liveness in Timed Systems

Summary

The previous chapter was devoted to the study of liveness on untimed net systems. The present chapter faces the same study on timed net systems. The framework of the study is given again by the class of mono-T-semiflow (MTS) systems. For the timed interpretation infinite servers semantics will be used as described in Chapter 1. As for untimed systems, in MTS timed systems it holds that deadlock-freeness is equivalent to liveness. A timed system is said to be live if the flow of its transitions is positive at the steady state. The concepts of critical-liveness and robust liveness are introduced. Critical-liveness stands for those systems that are live but a slight variation of their firing rates (internal speeds of transitions) can turn them into non live. On the other hand, robust-liveness applies to those systems that can be live for any value of their firing rates. The use of robust-liveness allows one to establish a connection between deadlock-freeness for untimed and timed systems.

Introduction

As for untimed net systems, liveness is a very desirable property for timed net systems. A first order approximation of the timed discrete systems is taken to introduce time in continuous Petri nets. Infinite servers semantics is considered in this chapter to define the flow through transitions. This way, a timed continuous Petri net system can be seen as a particular case of piecewise linear system (see Chapter 1 for details). The goal of this chapter is the study of liveness in timed continuous MTS Petri nets. The obtained results apply to any timed MTS Petri net that evolves through a transient state and reaches a steady state in which the marking of places and the flow through transitions are constant.

A timed system is said to be live when the flow of all its transitions is strictly positive at the steady state[JRS02, JRS]. In a similar way to untimed systems, in MTS timed systems deadlock-freeness is equivalent to liveness. Thus, in the steady state either all transitions have positive flow or all transitions have null flow. A timed net is the junction of a Petri net and a set of parameters, the firing rates of the transitions ($\boldsymbol{\lambda}$ according to Chapter 1). Timed structural liveness is a property that holds for those "timed nets" for which an initial marking exists such that the system is live. Clearly, structural liveness is a necessary condition for liveness. The set of firing rates that makes a given net structurally live strongly depends on the net structure. This chapter shows how to compute the set of firing rates that makes a net structurally live. Such a set is denoted as $\boldsymbol{\Lambda}$. If the firing rate of a net is a border point of its $\boldsymbol{\Lambda}$, then the system is said to be critically live: Any small variation of its firing rates may cause the system to become non structurally live. The opposite concept to critical liveness is robust liveness. A net is robust live when it is structurally live for any firing rate.

Notice that if a timed system evolves to a deadlock, then the same system seen as untimed could be driven to a deadlock by following the same evolution. This fact implies that (structural) liveness of the timed system is a necessary condition for (structural) liveness of the untimed system. This relationship between untimed and timed systems allows one to improve some necessary liveness conditions obtained in the previous chapter.

Section 4.1 shows that liveness of a given timed discrete system is in general not preserved by the timed continuous system. This phenomenon already appeared when considering untimed systems (see Chapter 3). Deadlock-freeness and liveness in timed systems are defined and some results are given in Section 4.2. Structural liveness is studied in detail in Section 4.3. The concepts of critical liveness and robust liveness are developed in Section 4.4 and Section 4.5 respectively. Section 4.6 derives a necessary condition for liveness of untimed systems from the results obtained for timed systems.

4.1 No liveness preservation in timed systems

As an example of the "mismatches" among properties of timed models, it can be pointed out that the addition of the *infinite servers semantics* time interpretation may allow the timed continuous model to have infinite behavior (deadlock-freeness), while a "similar" timing in the discrete system leads to a deadlock. Looking at the system in Figure 4.1(a) as continuous with $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$, it can be checked that it is live and the flow through transitions t_1 and t_2 is always the same. However, if one looks at the system as discrete and considers a classical markovian time interpretation, the stochastic system will arrive to a deadlock marking with probability "1" (this is a particular case of the classical "gambler's ruin problem"). If $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$ the mean time for deadlock is a quadratic function of k (see [SR02] for more details).

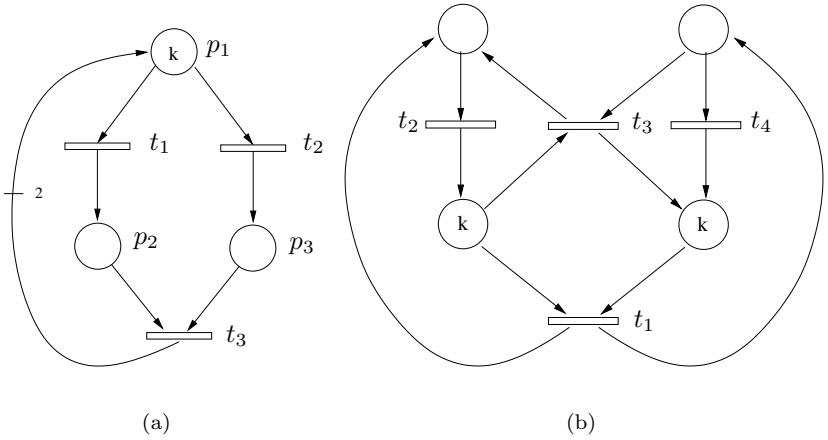


Figure 4.1: Timed systems whose discrete behaviour is different from the continuous one.

Liveness of a transition can also be affected when considering a system as discrete or as continuous. Figure 4.1(b) shows a non-MTS system (it is consistent and conservative, but has two T-semiflows), that considered as discrete is live (thus deadlock-free), but for which a deterministic timing of transitions with t_4 faster than t_2 (i.e., $\theta_4 < \theta_2$) makes t_3 non-live (in fact, it starves). Nevertheless, considering the model as continuous, it is live both for the untimed and the timed interpretations.

4.2 Deadlock-freeness and liveness in timed systems

In order to define deadlock-freeness and liveness of timed net systems the flow of the transitions in the steady state is considered. Observe that in the steady state

$\dot{\mathbf{m}}(\tau) = 0$, and so, from the state equation, $\mathbf{C} \cdot \mathbf{f}_{ss} = \mathbf{0}$ where $\mathbf{f}_{ss}$ is the flow vector of the timed system in the steady state, $\mathbf{f}_{ss} = \lim_{\tau \to \infty} \mathbf{f}(\tau)$. Since $\mathbf{f}_{ss} \geq \mathbf{0}$, the flow in the steady state is proportional to *the minimal* T-semiflow. The marking at the steady state will be denoted as $\mathbf{m_{ss}}$.

Definition 4.1. Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ be a timed continuous PN system.

- $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ is *timed-deadlock-free* iff $\mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) \neq \mathbf{0}$
- $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ is *timed-live* iff $\mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) > \mathbf{0}$
- $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is *str. timed-deadlock-free* iff there exists $\mathbf{m_0}$ such that $\mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) \neq \mathbf{0}$
- $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is *str. timed-live* iff there exists $\mathbf{m_0}$ such that $\mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) > \mathbf{0}$

As in untimed nets, str. timed-deadlock-freeness is a necessary condition for timed-liveness: If a timed MTS system is not timed-live (timed-deadlock-free), it can be concluded that, seen as untimed, the system is non lim-live (lim-deadlock-free) since the evolution of the timed system just gives a particular trajectory, i.e., a firing sequence, that can be fired in the untimed system. Therefore lim-liveness (lim-deadlock-freeness) is a sufficient condition for timed-liveness (timed-deadlock-freeness). The reverse is not true (Figure 4.1(a) for $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$). Analogously, str. lim-liveness is a sufficient condition for str. timed-liveness. Relationships among liveness definitions are depicted in Figure 4.2.

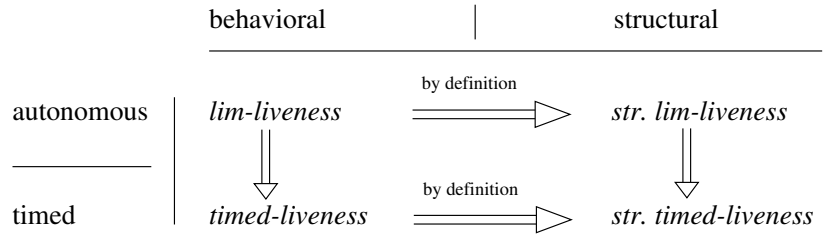


Figure 4.2: Relationships among liveness definitions for continuous MTS models.

Notice that in the steady state the flow through transitions of a MTS system, $\mathbf{f}_{ss}$, is proportional to the only T-semiflow of the net, or equivalently to a normalized vector of visit ratios, $\mathbf{v}^{(1)}$ where $\mathbf{v}^{(1)}[t_1] = 1$. Hence, the marking in the steady state, that will be denoted as $\mathbf{m_{ss}}$, verifies:

$$\forall\ t\ \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\mathbf{m_{ss}}[p]}{\mathbf{Pre}[p,t]} \right\} = \mathbf{f}_{ss}[t] = k \cdot \mathbf{v}^{(1)}[t] \tag{4.1}$$

Clearly, a MTS system under infinite servers semantics, $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$, is timed-live iff $k > 0$.

4.3 Structural timed-liveness

The vector of firing speeds $\boldsymbol{\lambda}$ plays a crucial role in the evolution to the steady state. As the system in Figure 4.1(a) shows, even str. non-lim-live systems can be saved from deadlocking by choosing an adequate $\boldsymbol{\lambda}$. It can be seen that for any strictly positive initial marking it is always possible to find a $\boldsymbol{\lambda}$ that makes the system timed-live. One idea is to chose a $\boldsymbol{\lambda}$ that avoids any transient state, thus making the initial marking equal to the marking in the steady state and therefore avoiding a deadlock.

Proposition 4.2. Given a MTS net, $\mathcal{N}$, for every initial marking $\mathbf{m_0} > \mathbf{0}$, there exists $\boldsymbol{\lambda}$ such that $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ is timed-live.

Proof. Let us define $\boldsymbol{\lambda}[t_i] = \frac{\mathbf{v}^{(1)}[t_i]}{\text{enab}(t_i, \mathbf{m_0})}$ where $\mathbf{v}^{(1)}$ is the vector of visit ratios normalized for t_1. Since $\mathbf{f}[t_i](\tau = 0) = \boldsymbol{\lambda}[t_i] \cdot \text{enab}(t_i, \mathbf{m_0})$ it is immediate to verify that $\mathbf{f}(\tau = 0) = \mathbf{v}^{(1)} = \mathbf{f}_{ss} > \mathbf{0}$ □

For example, the continuous system in Figure 4.3 is not lim-live as untimed. However, defining $\boldsymbol{\lambda} = (1\ 1\ \lambda_3)$ the timed version never deadlocks.

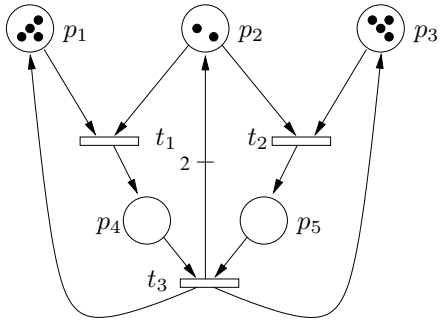


Figure 4.3: A non lim-live untimed system that never deadlocks as timed with $\boldsymbol{\lambda} = (1\ 1\ \lambda_3)$.

Another interesting problem is to determine which continuous timed-nets are str. timed-live (i.e., given $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$, $\exists\ \mathbf{m_0}$ such that $\mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) > \mathbf{0}$?).

Proposition 4.3. $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live iff $\mathbf{m}$ defined as

$$\mathbf{m}[p] = \max_{t \in p^\bullet} \left\{ \frac{\mathbf{Pre}[p,t] \cdot \mathbf{v}^{(1)}[t]}{\boldsymbol{\lambda}[t]} \right\}$$

is a steady state marking for $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$.

Proof.
($\Leftarrow$)
If $\mathbf{m}$ is a steady state marking, then $\mathbf{f}_{ss} = \mathbf{v}^{(1)}$ and $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live.

($\Rightarrow$)
There exists $\mathbf{m_0}$ such that $\mathbf{f}_{ss}[t_1](\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) = k > 0$. Let $\mathbf{m}_{ss}$ be the steady state marking associated to $\mathbf{m_0}$. Let us define $\boldsymbol{\mu} = \frac{\mathbf{m}_{ss}}{k}$. Clearly $\mathbf{f}_{ss}[t_1](\mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 1$. Then, $\forall\, p\ \ \boldsymbol{\mu}[p] \geq \max_{t \in p^\bullet}\{\frac{\mathbf{Pre}[p,t] \cdot \mathbf{v}^{(1)}[t]}{\boldsymbol{\lambda}[t]}\}$. The components of $\boldsymbol{\mu}$ being strictly greater can be made equal without modifying $\mathbf{f}_{ss}$ and the result is $\mathbf{m}$, as defined in the statement. □

Let us apply Proposition 4.3 to the net in Figure 3.3 with $\boldsymbol{\lambda} = (4\ 1\ 1)$. The vector of visit ratios of the net is $\mathbf{v}^{(1)} = \mathbf{1}$, and so the marking defined by the statement of Proposition 4.3 is $\mathbf{m} = (1\ 1\ 1\ 1)$. This marking is not a steady state marking since it does not verify Equation (4.1). Therefore, the timed net with $\boldsymbol{\lambda} = (4\ 1\ 1)$ is not str. timed-live.

4.3.1 Characterization of the $\boldsymbol{\Lambda}_{\mathcal{N}}$ set

Given $\mathcal{N}$, an essential problem lies in determining the set of firing speed vectors for which $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live. In other words, the goal is to compute a set defined as follows:

Definition 4.4. $\boldsymbol{\Lambda}_{\mathcal{N}} = \{\boldsymbol{\lambda} \mid \langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live $\}$.

It has been seen that if $\mathcal{N}$ is str. lim-live then for any $\boldsymbol{\lambda}$, $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live (recall Figure 4.2). Hence for str. lim-live nets $\boldsymbol{\Lambda}_{\mathcal{N}}$ will be equal to all positive real vectors ($\boldsymbol{\Lambda}_{\mathcal{N}} = (\mathbb{R}^+)^{|T|}$).

Let us show that the computation of $\boldsymbol{\Lambda}_{\mathcal{N}}$ can be simplified by considering separately each coupled conflict set.

Let us suppose that $\mathcal{N}$ has q coupled conflict sets, $CCS_1 \ldots CCS_q$ with $|CCS_1| = n_1, \ldots, |CCS_q| = n_q$, $|{}^\bullet CCS_1| = s_1, \ldots, |{}^\bullet CCS_q| = s_q$, and that transitions and places are sorted according to the coupled conflict they belong to: $t_{1,1} \ldots t_{1,n_1}$, $\ldots, t_{q,1} \ldots t_{q,n_q}$ and $p_{1,1} \ldots p_{1,s_1}, \ldots, p_{q,1} \ldots p_{q,s_q}$. Let us define $\boldsymbol{\Lambda}_{CCS_j}$ as a set of vectors associated to the coupled conflict set CCS_j in the following way:

Definition 4.5. $\boldsymbol{\Lambda}_{CCS_j} = \{\boldsymbol{\lambda}_j \mid \boldsymbol{\lambda}_j \in (\mathbb{R}^+)^{n_j}$ and $\exists\ \mathbf{m} \in (\mathbb{R}^+)^{s_j}$ such that $\forall\ t \in CCS_j\ \ \mathbf{v}^{(1)}[t] = \boldsymbol{\lambda}_j[t] \cdot \mathrm{enab}(t, \mathbf{m})\}$

The following Theorem states that $\boldsymbol{\Lambda}_{\mathcal{N}}$ can be expressed as the cartesian product of all the $\boldsymbol{\Lambda}_{CCS}$ of the net.

Theorem 4.6. $\boldsymbol{\Lambda}_{\mathcal{N}} = \{(\lambda_{1,1}, \ldots, \lambda_{q,n_q}) \mid (\lambda_{i,1}, \ldots, \lambda_{i,n_i}) \in \boldsymbol{\Lambda}_{CCS_i}\}$

Proof.

- $\boldsymbol{\Lambda}_{\mathcal{N}} \subseteq \{(\lambda_{1,1}, \ldots, \lambda_{q,n_q}) \mid (\lambda_{i,1}, \ldots, \lambda_{i,n_i}) \in \boldsymbol{\Lambda}_{CCS_i}\}$
  If $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$, there exists $\mathbf{m_0}$ such that $\mathbf{f}_{ss}[t_1](\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0}) = l > \mathbf{0}$. Let $\mathbf{m_{ss}}$ be its associated steady state marking. Then $\forall\ i = 1 \ldots q$, $\boldsymbol{\lambda}_i = \{\lambda_{i,1} \ldots \lambda_{i,n_i}\} \in \boldsymbol{\Lambda}_{CCS_i}$ with $\mathbf{m}_{i,j} = \frac{\mathbf{m_{ss}}_{i,j}}{l}$.

- $\boldsymbol{\Lambda}_{\mathcal{N}} \supseteq \{(\lambda_{1,1},\ldots,\lambda_{q,n_q}) \mid (\lambda_{i,1},\ldots,\lambda_{i,n_i}) \in \boldsymbol{\Lambda}_{CCS_i}\}$
  If $\boldsymbol{\lambda} \in \{(\lambda_{1,1},\ldots,\lambda_{q,n_q}) \mid (\boldsymbol{\lambda}_{i,1},\ldots,\boldsymbol{\lambda}_{i,n_i}) \in \boldsymbol{\Lambda}_{CCS_i}\}$ with the marking $m_{1,1}\ldots m_{1,s_1}, \quad m_{2,1}\ldots m_{2,s_2}, m_{q,1}\ldots m_{q,s_q}$, then using this marking as $\mathbf{m_0}$, one has $\mathbf{f}_{ss}(\mathcal{N},\boldsymbol{\lambda},\mathbf{m_0}) = \mathbf{v}^{(1)} > \mathbf{0}$. Therefore $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$

□

According to Proposition 4.2, for every positive initial marking, $\mathbf{m_0}$, if $\boldsymbol{\lambda}$ is adequately chosen, the system reaches a non-dead steady state. In fact the initial marking is also the marking at the steady state. Clearly, in that case $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$. In the proof of Proposition 4.2 $\boldsymbol{\lambda}[i]$ is computed just by dividing the visit ratio of t_i by its enabling degree at $\mathbf{m_0}$. If this operation is executed on every positive marking one obtains all the $\boldsymbol{\lambda}$ vectors contained in $\boldsymbol{\Lambda}_{\mathcal{N}}$, since all positive markings represent all possible steady states. The result of applying this reasoning to a CCS yields the following expression for $\boldsymbol{\Lambda}_{CCS}$.

Proposition 4.7. Let CCS be a coupled conflict set such that $|CCS| = n$ and $|{}^{\bullet}CCS| = s$. Then:

$$\boldsymbol{\Lambda}_{CCS} = \{\boldsymbol{\lambda} | \boldsymbol{\lambda}[t_i] = \mathbf{v}^{(1)}[t_i] \cdot \max_{p_j \in {}^{\bullet}t_i} \{\alpha_j \cdot \mathbf{Pre}[p_j,t_i]\}, \boldsymbol{\alpha} \in (\mathbb{R}^+)^s\}$$

Proof. Let $\mathbf{m} > \mathbf{0}$, for every $t_i \in CCS$. Let us define $\boldsymbol{\lambda}$ as $\boldsymbol{\lambda}[t_i] = \frac{\mathbf{v}^{(1)}[t_i]}{\mathrm{enab}(t_i,\mathbf{m})}$. Using Proposition 4.2 it is known that $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{CCS}$. One can rewrite $\boldsymbol{\lambda}$ as $\boldsymbol{\lambda}[t_i] = \frac{\mathbf{v}[t_i]}{\min_{p_j \in {}^{\bullet}t_i}\{\frac{\mathbf{m}[p_j]}{\mathbf{Pre}[p_j,t_i]}\}} = \mathbf{v}[t_i] \cdot \max_{p_j \in {}^{\bullet}t_i}\{\frac{\mathbf{Pre}[p_j,t_i]}{\mathbf{m}[p_j]}\}$. Defining $\alpha_j = \frac{1}{\mathbf{m}[p_j]}$ one obtains $\boldsymbol{\lambda}[t_i] = \mathbf{v}^{(1)}[t_i] \cdot \max_{p_j \in {}^{\bullet}t_i}\{\alpha_j \cdot \mathbf{Pre}[p_j,t_i]\}$ with $\boldsymbol{\alpha} \in (\mathbb{R}^+)^s$ □

Let us consider the net depicted in Figure 3.3. Its vector of visit ratios is $\mathbf{v}^{(1)} = (1\ 1\ 1)$ and it has two CCSs: CCS_{t_1,t_2} and CCS_{t_3}. Applying Proposition 4.7 the following sets are computed: $\boldsymbol{\Lambda}_{CCS_{t_1,t_2}} = \{(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) | \boldsymbol{\lambda}_1 = \alpha_1, \boldsymbol{\lambda}_2 = \max\{\alpha_1 \cdot 2, \alpha_2\}, \alpha_1 > 0, \alpha_2 > 0\} = \{(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) | \boldsymbol{\lambda}_2 \geq 2 \cdot \boldsymbol{\lambda}_1 > 0\}$ and $\boldsymbol{\Lambda}_{CCS_{t_3}} = \{(\boldsymbol{\lambda}_3) | \boldsymbol{\lambda}_3 = \alpha_3, \alpha_3 > 0\}$. A direct application of Theorem 4.6 on these sets allows one to obtain the set $\boldsymbol{\Lambda}_{\mathcal{N}} = \{(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\boldsymbol{\lambda}_3) | \boldsymbol{\lambda}_2 \geq 2 \cdot \boldsymbol{\lambda}_1 > 0, \boldsymbol{\lambda}_3 > 0\}$, i.e., the set of firing speeds for which the system allows a non dead steady state.

4.3.2 Restrictive places

The expression in Proposition 4.7 describes the set of internal speeds of the transitions in a CCS that have to be considered in order to avoid the system to deadlock. That is, if the internal speeds of the transitions are in $\boldsymbol{\Lambda}_{CCS}$, there exists a marking at which the transitions are fired proportionally to the vector of visit ratios. This means that at that marking every transition is adequately enabled by at least one of its input places.

Definition 4.8. Given a steady state marking, $\mathbf{m}$, a place p is said to be a *restrictive* place for its output transition t if the enabling degree of t at $\mathbf{m}$ is defined by the marking of p, i.e., $\text{enab}(t, \mathbf{m}) = \mathbf{m}[p]/\mathbf{Pre}[p, t]$.

For a given $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$, it turns out that in the steady state a place can be restrictive only for some of its output transitions. The set of transitions for which a place can be restrictive depends on the structure of the net and on the internal speeds of the transitions. This fact can be interpreted in the following way: in the steady state every transition is "demanding" a minimum quantity of marking to their input places in order to have a positive throughput. From another point of view, this means that places have to supply enough fluid to their output transitions. Hence, in the steady state, a place can be restrictive only for the output transition(s) that is(are) demanding the greatest amount of fluid.

Let us assume that the system $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ reaches a steady state at which the throughput is $\mathbf{f}_{ss}$. Thus, the marking of a given place p has to be big enough to allow its output transitions to fire according to $\mathbf{f}_{ss}$. Considering all the output transitions of a place p, its marking in the steady state, $\mathbf{m_{ss}}[p]$, has to fulfill:

$$\mathbf{m_{ss}}[p] \geq \max_{t_i \in p^\bullet} \left\{ \frac{\mathbf{Pre}[p, t_i]}{\boldsymbol{\lambda}_i} \cdot \mathbf{f}_{ss}[t_i] \right\} \tag{4.2}$$

This equation can be directly obtained from Equation (4.1). Given a place p, Equation (4.2) allows one to compute which of its output transitions is demanding the greatest marking in the steady state. And so, it is possible to deduce the transition(s) for which the place p can be restrictive. Since only MTS systems are being considered, $\mathbf{f}_{ss}$ is proportional to the only T-semiflow of the system. Therefore, taking into account Equation (4.2), the computation of the transition(s) for which a place p can be restrictive only depends on the T-semiflow of the system and on the vector of internal speeds $\boldsymbol{\lambda}$, i.e., it does not depend on the initial marking.

Proposition 4.9. The set of transitions for which a place p can be restrictive is given by:

$$\left\{ t_j \in p^\bullet \mid \frac{\mathbf{Pre}[p, t_j]}{\boldsymbol{\lambda}_j} \cdot \mathbf{v}^{(1)}[t_j] = \max_{t_i \in p^\bullet} \left\{ \frac{\mathbf{Pre}[p, t_i]}{\boldsymbol{\lambda}_i} \cdot \mathbf{v}^{(1)}[t_i] \right\} \right\} \tag{4.3}$$

where $\mathbf{v}^{(1)}$ is the vector of visit ratios normalized for transition t_1.

According to Proposition 4.9, a place p can be restrictive for more than one transition only in the case that several of its output transitions are demanding exactly the same marking to place p. Notice that every transition has an input restrictive place. However not every place has to be a restrictive place for one of its output transitions.

Let us assume that given a $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ the only possible restrictive place for a transition t is the place p. Thus, in a non dead steady state, place p has the *responsibility* of adequately enabling transition t. The task of enabling transition t will be impossible for place p if it is an implicit place [RTS98]. Therefore, if the system is desired to reach a non dead steady state, it must be avoided that the set of restrictive places of a transition is implicit.

In order to illustrate these reasonings, let us consider a CCS composed of two places and two transitions whose normalized visit ratios are $\mathbf{v}_1$ and $\mathbf{v}_2$, see Figure 4.4(a). According to Proposition 4.7, the $\boldsymbol{\Lambda}_{CCS}$ associated to that CCS is $\boldsymbol{\Lambda}_{CCS} = (\lambda_1, \lambda_2) = \{(\mathbf{v}_1 \cdot \max\{\alpha_1 \cdot a, \alpha_2 \cdot c\}, \mathbf{v}_2 \cdot \max\{\alpha_1 \cdot b, \alpha_2 \cdot d\}) \mid \boldsymbol{\alpha} \in (\mathbb{R}^+)^2\}$. Through some algebraic operations it can be seen that the region in the two dimensional plane associated to this $\boldsymbol{\Lambda}_{CCS}$ can be written as: $\boldsymbol{\Lambda}_{CCS} = (\lambda_1, \lambda_2) = \{\beta_1 \cdot (\mathbf{v}_1 \cdot a, \mathbf{v}_2 \cdot b) + \beta_2 \cdot (\mathbf{v}_1 \cdot c, \mathbf{v}_2 \cdot d) \mid \boldsymbol{\beta} \in (\mathbb{R}^+)^2\}$. That is, the set $\boldsymbol{\Lambda}_{CCS}$ can be seen as a two dimensional cone.

Let us assume, without loss of generality, that $\frac{d}{c} \geq \frac{b}{a}$. The slopes of the upper and lower edges of the cone are $\frac{d}{c}$ and $\frac{b}{a}$ respectively. Figure 4.4(b) depicts the region associated to $\boldsymbol{\Lambda}_{CCS}$.

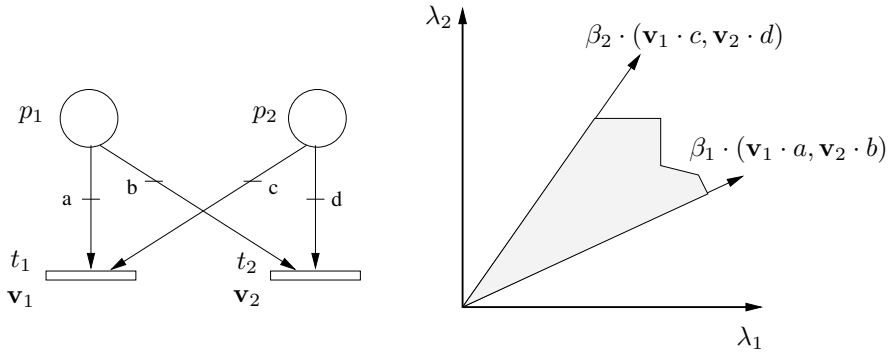


Figure 4.4: (a) A simple CCS. (b) The $\boldsymbol{\Lambda}_{CCS}$ set associated to the CCS assuming $\frac{d}{c} \geq \frac{b}{a}$.

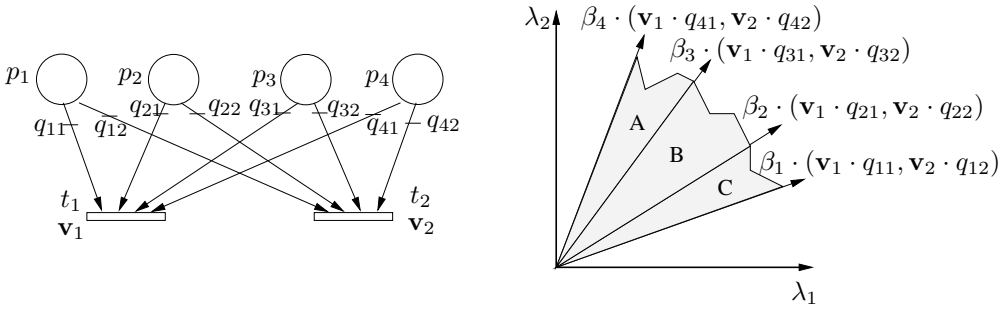
Only those speeds, (λ_1, λ_2), in the cone allow a non dead steady state. Assuming that this CCS is part of a MTS, $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$, and that a non dead steady state is reached, the following results based on Equation 4.3 hold:

- If (λ_1, λ_2) is not a border point of the cone $\boldsymbol{\Lambda}_{CCS}$, then p_1 is the restrictive place of t_1 and p_2 is the restrictive place of t_2. Therefore, if any of the places is

implicit the system will deadlock.

- If (λ_1, λ_2) is a point in the upper edge of the cone then p_1 is restrictive for $t1$ and p_2 is restrictive for both transitions. If the system is wanted to reach a non dead steady state it is necessary that place p_2 is not implicit.
- If (λ_1, λ_2) is a point in the lower edge of the cone then p_1 is restrictive for both transitions and p_2 is restrictive for t_2. In this case if p_1 is implicit the system will deadlock.

Its not difficult to extend the previous results to a more complex CCS consisting of several input places and two transitions. The CCS in Figure 4.5(a) has 4 input places and two transitions. The region containing the set $\mathbf{\Lambda}_{CCS}$ has the shape of a cone. See Figure 4.5(b) for the representation of $\mathbf{\Lambda}_{CCS}$ assuming that $\frac{q_{42}}{q_{41}} \geq \frac{q_{32}}{q_{31}} \geq \frac{q_{22}}{q_{21}} \geq \frac{q_{12}}{q_{11}}$.



(a) (b)

Figure 4.5: (a) A CCS with four places and two transitions. (b) The $\mathbf{\Lambda}_{CCS}$ set associated to the CCS assuming that $q_{42}/q_{41} \geq q_{32}/q_{31} \geq q_{22}/q_{21} \geq q_{12}/q_{11}$.

Three cones can be distinguished in the interior of $\mathbf{\Lambda}_{CCS}$. The set of restrictive places depends on the cone to which (λ_1, λ_2) belongs. If (λ_1, λ_2) belongs to the cone A, the possible restrictive places for t_1 are p_1, p_2, p_3 and for t_2 the only possible restrictive place is p_4. Hence, in this case if p_4 is implicit the system will deadlock. A deadlock will also be reached if p_1, p_2 and p_3 are implicit. For a (λ_1, λ_2) in the cone B, the restrictive places for transition t_1 are p_1 and p_2, and for transition t_2 the restrictive places are p_3 and p_4. If (λ_1, λ_2) is in cone C, the only restrictive place for transition t_1 is p_1 and the possible restrictive places for transition t_2 are p_2, p_3 and p_4. Finally, notice that in the edges of the cones it turns out that a single place can be restrictive for both transitions. For example for a (λ_1, λ_2) in the upper edge of the cone, place p_4 is restrictive for both transitions.

4.4 Critical timed-liveness

It has been seen that those $\boldsymbol{\lambda}$ vectors not included in $\boldsymbol{\Lambda}_{\mathcal{N}}$ do not allow a MTS system to reach a steady state with throughput greater than zero. Although $\boldsymbol{\Lambda}_{\mathcal{N}}$ is never an empty set (for every positive initial marking there exists $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$, Proposition 4.2), its "size" can be much smaller than desired. For example it is not desirable to use a vector of $\boldsymbol{\Lambda}_{\mathcal{N}}$ such that a minimum change in one of its components puts the vector out of $\boldsymbol{\Lambda}_{\mathcal{N}}$. It would mean that a small variation in the firing speed of one transition can kill the system. Hence, a new concept is needed to define whether a system can be "robust enough" to bear irregularities and variations happening in the real world.

Definition 4.10. $(\mathcal{N}, \boldsymbol{\lambda})$ is critically str. timed-live iff $\boldsymbol{\lambda}$ is a border point of $\boldsymbol{\Lambda}_{\mathcal{N}}$

In some cases the net structure can reduce dramatically the dimension of $\boldsymbol{\Lambda}_{\mathcal{N}}$. For every coupled conflict with n transitions $\boldsymbol{\Lambda}_{CCS}$ is contained in $(\mathbb{R}^+)^{|n|}$. Therefore the maximum dimension that a given region of the $\boldsymbol{\Lambda}_{CCS}$ set can have is n. Apart from this constraint, the effective dimension of $\boldsymbol{\Lambda}_{CCS}$ is also limited by the number of input places of the coupled conflict set, since $\boldsymbol{\Lambda}_{CCS}$ is generated by as many independent variables as input places (see Proposition 4.7). Therefore, the following Proposition holds:

Proposition 4.11. Given a coupled conflict set CCS, the maximum dimension of any region of $\boldsymbol{\Lambda}_{CCS}$ is bounded by the number of transitions in the CCS, $|CCS|$, and the number of input places of the CCS, $|{}^{\bullet}CCS|$.

Since $\boldsymbol{\Lambda}_{\mathcal{N}}$ is the cartesian product of all the $\boldsymbol{\Lambda}_{CCS}$ of the net (Theorem 4.6), if the net has a coupled conflict set CCS with less input places than transitions ($|{}^{\bullet}CCS| < |CCS|$), every region in $\boldsymbol{\Lambda}_{\mathcal{N}}$ will have a smaller dimension than the number of transition of the net. That is, every point in $\boldsymbol{\Lambda}_{\mathcal{N}}$ will be a border point. From this reasoning, Proposition 4.12 is derived.

Proposition 4.12. Given a net $\mathcal{N}$, if there exists a CCS such that $|{}^{\bullet}CCS| < |CCS|$ then for every $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$ $(\mathcal{N}, \boldsymbol{\lambda})$ is critically str. timed-live.

For example, the CCS composed of $\{t_1, t_2\}$ in Figure 4.1(a) has only one input place, p_1. Hence, if $\boldsymbol{\lambda} \notin \boldsymbol{\Lambda}_{\mathcal{N}}$ then for every initial marking the system will finally deadlock, and if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\mathcal{N}}$ then $(\mathcal{N}, \boldsymbol{\lambda})$ is critically str. timed-live.

The above means that in practice, all the coupled conflicts sets should have at least as many input places as transitions, otherwise the system will die or will remain in a critical timed-live state.

4.5 Robust timed-liveness

Going on with critical str. timed-liveness, one could ask which features should be required to a system in order to be "robust", i.e., arbitrary variations in $\boldsymbol{\lambda} > 0$ do not cause $\boldsymbol{\lambda}$ to be out of $\boldsymbol{\Lambda}_{\mathcal{N}}$. In other words, the interest lies on those net structures that for any $\boldsymbol{\lambda}$ allow a non-dead steady state.

A place p is said to be *choice-free (CF)* iff $|p^\bullet| = 1$, i.e., p has a single output transition. It will be said that a transition *owns* its (input) CF places.

Theorem 4.13. Let $\mathcal{N}$ be a MTS net, $\forall\ \boldsymbol{\lambda} > 0$ $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is str. timed-live iff every transition owns at least one CF place.

Proof.
($\Leftarrow$)
Let $\boldsymbol{\lambda} \in (\mathbb{R}^+)^{|T|}$. Let $P' = \{p_1, \ldots, p_k\}$ be a minimal set of CF places such that $(P')^\bullet = T$. For every $p_i \in P'$ let $t_i = p_i{}^\bullet$, and define $\mathbf{m}[p_i] = \frac{\mathbf{Pre}[p_j,t_i]\cdot \mathbf{v}^{(1)}[t_i]}{\boldsymbol{\lambda}[t_i]}$ if $p_i \in P'$, and $\mathbf{m}[p_i] = \max_{t_j \in p_i{}^\bullet}\{\frac{\mathbf{Pre}[p_i,t_j]\cdot \mathbf{v}^{(1)}[t_j]}{\boldsymbol{\lambda}[t_j]}\}$ otherwise. One has that for every $t_i \in T$ $\boldsymbol{\lambda}[t_i] \cdot \min_{p_j \in {}^\bullet t_i}\{\frac{\mathbf{m}[p_j]}{\mathbf{Pre}[p_j,t_i]}\} = \mathbf{v}^{(1)}[t_i]$. Therefore $\mathbf{m}$ is a steady state marking.
($\Rightarrow$)
Let $\boldsymbol{\Lambda}_{\mathcal{N}} = (\mathbb{R}^+)^{|T|}$ and assume $\exists\ t_1$ without CF places. Then $\forall\ p \in {}^\bullet t_1\ |p^\bullet| > 1$. Let $\{t_1, \ldots, t_k\} = ({}^\bullet t_1)^\bullet \subseteq CCS(t_1)$. According to Theorem 4.6 $\boldsymbol{\Lambda}_{\mathcal{N}}$ is the product of $\boldsymbol{\Lambda}_{CCS(t_i)}$, so $\boldsymbol{\Lambda}_{CCS(t_1)} = (\mathbb{R}^+)^{|CCS(t_1)|}$. Applying Proposition 4.7, for any $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{CCS(t_1)}$, a certain $\boldsymbol{\alpha} \in (\mathbb{R}^+)^{|{}^\bullet CCS(t_1)|}$ exists verifying $\boldsymbol{\lambda}[t_i] = \mathbf{v}^{(1)}[t_i] \cdot \max_{\{p_j \in {}^\bullet t_i\}}\{\alpha_j \cdot \mathbf{Pre}[p_j, t_i]\}$ for every $1 \leq i \leq k$. Therefore, $\boldsymbol{\lambda}[t_1] = \mathbf{v}^{(1)}[t_1] \cdot \alpha_h \cdot \mathbf{Pre}[p_h, t_1]$ for a certain $p_h \in {}^\bullet t_1$. Since $|p_h{}^\bullet| > 1$, another transition $t_j \in \{t_2, \ldots, t_k\}$ exists such that $t_j \in p_h{}^\bullet$, and so $\boldsymbol{\lambda}[t_j] \geq \mathbf{v}^{(1)}[t_j] \cdot \alpha_h \cdot \mathbf{Pre}[p_h, t_j]$. Hence, $\frac{\boldsymbol{\lambda}[t_1]}{\mathbf{v}^{(1)}[t_1]\cdot \mathbf{Pre}[p_h,t_1]} = \alpha_h \leq \frac{\boldsymbol{\lambda}[t_j]}{\mathbf{v}^{(1)}[t_j]\cdot \mathbf{Pre}[p_h,t_j]} \leq \max_{\{j\in\{2\ldots k\}\}}\left\{\frac{\boldsymbol{\lambda}[t_j]}{\mathbf{v}^{(1)}[t_j]\cdot \mathbf{Pre}[p_h,t_j]}\right\}$. Therefore, $\boldsymbol{\lambda}[t_1] \leq \mathbf{v}^{(1)}[t_1] \cdot \mathbf{Pre}[p_h, t_1] \cdot \max_{\{j\in\{2\ldots k\}\}}\left\{\frac{\boldsymbol{\lambda}[t_j]}{\mathbf{v}^{(1)}[t_j]\cdot \mathbf{Pre}[p_h,t_j]}\right\}$. That is, $\boldsymbol{\lambda}[t_1]$ is bounded by a value that depends on $\boldsymbol{\lambda}[t_2], \ldots, \boldsymbol{\lambda}[t_k]$. Contradiction. □

From a different point of view, Theorem 4.13 states that the transitions can be enabled independently iff every transition owns a CF place. Observe that this condition does not guarantee that the system will always reach a non-dead steady state for every initial marking. For example in Figure 4.3, every transition owns a CF place (and it is not a CF net). However, with that initial marking, choosing $\boldsymbol{\lambda} = (2\ 1\ 1)$ the system cannot reach a live steady state. This happens because the enabling degree of t_1 and t_2 is always the same, since p_1 and p_3 are implicit places [RTS98], so they can be removed without changing the possible behaviors (trajectories for the timed case) of the system. From Theorem 4.13 it can be inferred that for those nets that have a

transition without CF places there exists a $\boldsymbol{\lambda}$ for which the timed system deadlocks independently of the initial marking. Transitions $\{t_1, t_2, t_3\}$ in Figure 4.6 do not own CF places. For $\boldsymbol{\lambda} = (1\ 1\ 2\ 1\ 1\ 1)$, $\boldsymbol{\lambda} \notin \boldsymbol{\Lambda}_{\mathcal{N}}$, the system will evolve to a deadlock, no matter which the initial marking is.

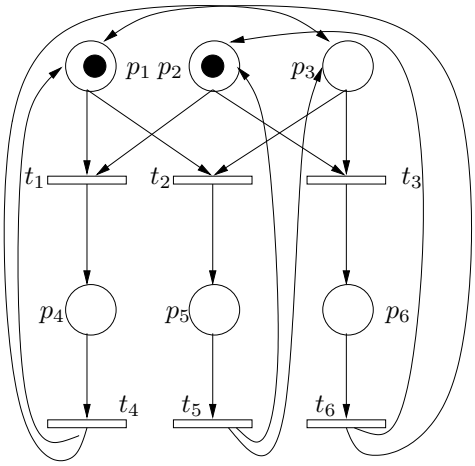


Figure 4.6: Transitions t_1, t_2 and t_3 do not own CF places. For $\boldsymbol{\lambda} = (1\ 1\ 2\ 1\ 1\ 1)$ no steady state with positive throughput is possible.

4.6 Coming back to structural liveness in untimed systems

According to Theorem 4.13, if $\mathcal{N}$ has a transition without CF places, a $\boldsymbol{\lambda}$ exists such that $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is not str. timed-live. Therefore $\mathcal{N}$ is not str. lim-live, since str. timed-liveness is a necessary condition for str. lim-liveness.

Theorem 4.14. Let $\mathcal{N}$ be a MTS net. If $\mathcal{N}$ is str. lim-live then every transition owns at least one CF place.

The system shown in Figure 4.6 is not lim-live according to Theorem 4.14, since there are three transitions, t_1, t_2 and t_3 that do not own a CF place. In this case, the firing of a sequence that corresponds to vector $\sigma = (0\ 1\ 0\ 0\ 1\ 0)$ leads to marking $\mathbf{m} = (0\ 2\ 0\ 0\ 0\ 0)$, where the system lim-deadlocks. Notice that in consistent continuous systems in which every transition can be fired at least once, there do not exist spurious solutions of the state equation (Chapter 2), hence a sequence can be fired that reaches marking $\mathbf{m}$.

Transition t does not own a CF place iff all the input places of t are contained in the set of input places of the rest of the transitions. Thus, Theorem 4.14 can be rewritten as: If $\mathcal{N}$ is str. lim-live then for every t, ${}^\bullet t \not\subseteq {}^\bullet(T\backslash\{t\})$. Notice the similarity of this statement to that of Corollary 3.5. In fact, Theorem 4.14 and Corollary 3.5 express exactly the same condition if all the coupled conflict sets of the net have at most two transitions, but in general Theorem 4.14 has a greater decision power.

4.7 Conclusions

Liveness of timed systems is a very interesting property asking for positive flow through all transitions in the steady state. Liveness of a net system depends on its structure, the firing rates of its transitions and its initial marking. In general, all three elements have to be considered to evaluate the liveness of a given timed system.

As in untimed systems structural lim-liveness can been defined for timed systems. For such systems it is possible to obtain a necessary and sufficient condition for structural timed liveness (Proposition 4.3). A new concept, *critical timed-liveness*, has been introduced in this chapter. A critical timed-live system can reach a live steady state, however any little variation in one of its firing speeds will cause the system to deadlock. The set of firing speeds that allow the system to reach non–dead steady states has been characterized. In contrast to critical liveness, a system is said to be robust live iff it is structurally live for any firing speeds of its transitions. It has been obtained that a system is robust live iff every transition t owns at least one CF place, i.e., an input place whose only output transition is t.

The existing relationships between lim-liveness of untimed systems and liveness of timed systems allows one to apply some results obtained for timed systems (Theorem 4.13) to untimed systems (Theorem 4.14). Thus, improving some previous results (Corollary 3.5) for untimed systems.

Chapter 5

Steady State Performance Evaluation

Summary

A common way to measure the performance of a system is to compute its throughput. Coarsely speaking the throughput is the number of times a given action is executed per time unit. Thus, the greater the throughput the better the performance of the system. It is shown that, in general, the throughput of a continuous Petri net is not an upper bound for the throughput of the same system seen as discrete. It is also remarkable that the throughput of a continuous net system, in general, does not fulfill any monotonicity property. This chapter mainly focuses on the computation of throughput bounds for mono-T-semiflow continuous Petri net systems. For that purpose, a Branch & Bound algorithm is designed. The constraints of that algorithm can be relaxed in order to obtain a linear programming problem. The conditions under which the system always reaches the computed bounds are extracted. The results related to the computation of the bounds can be directly applied to a larger class of nets called *mono-T-semiflow reducible*.

Introduction

Since continuization implies removing a constraint (the integrality of the firing), it could be thought that in timed systems the throughput of the continuous system is an upper bound of the same system seen as discrete. However, this is not always the case [SR04]. The main goal of this chapter is to study the throughput bounds for the subclass of *mono-T-semiflow* (MTS) *Petri nets* (PN) [RJS02, JRS04].

For the timing interpretation in transitions, a semantics of infinite servers will be used [SR02]. Under this firing semantics the continuous Petri net system behaves as a piecewise linear system [Son81]. The throughput bounds refer to the behaviour of the system in the steady state, i.e., when its marking and flow through transitions are constant. There exist some constraints that have to be verified by the marking of the system at the steady state. These constraints are inferred from the fundamental state equation and the definition of flow of a transition under infinite servers semantics. Since the flow of a transition is proportional to its enabling degree, a minimum operator appears in the constraints. A way to deal with such minimum operator is to design a Branch & Bound algorithm. Unfortunately the complexity of the algorithm in the worst case is exponential with respect to the number of places. However, the constraints containing the minimum operator can be relaxed and substituted by a set of inequalities. The use of these inequalities allows one to obtain a linear programming problem to compute throughput bounds. Clearly, there are cases in which the bounds computed by the linear programming problem are less accurate than those computed by the Branch & Bound algorithm.

The computation of bounds by the linear programming problem can be seen as a search for a bottleneck in the net system. The bottleneck is represented by the slowest P-semiflow in the system. It is possible to establish the conditions under which the system can reach this kind of bounds. Furthermore, if the system fulfils some given conditions the bound will be reached for sure.

The structure of the chapter is the following: In Section 5.1 some interesting/unexpected behaviours related to the throughput of continuous systems are presented. These behaviours show that the throughput of a discrete system is not necessarily upper bounded by the throughput of its relaxed continuous system. Some non monotonic behaviours of the throughput of a continuous net system are illustrated through examples. In Section 5.2, some techniques to compute throughput bounds are described and applied to a manufacturing example. First, an algorithm based on a branch & bound technique is presented to compute upper throughput bounds. A very similar algorithm can be designed to compute lower throughput bounds. Then, it is shown how upper bounds (less tight in general) can be polynomially computed by means of a single linear programming problem. Conditions for reachability of the bounds computed by this last approach are given. Even if MTS systems generalize

a certain number of classical net subclasses, including conflicts and synchronizations, Section 5.3 introduces a larger class of nets, *mono-T-semiflow reducible systems*, to which previous results can be applied. The main feature of a mono-T-semiflow reducible system is that its visit ratio under infinite servers semantics does not depend on the initial marking and can be computed in polynomial time.

5.1 Remarkable behaviours of timed continuous systems

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

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, $\mathbf{v}^{(i)}[t_j]$, is the average number of times t_j is visited (fired) for each visit to (firing of) the reference transition t_i. Observe that $\mathbf{v}^{(i)}$ is a "normalization" ($\mathbf{v}^{(i)}[t_i] = 1$) of the flow vector in the steady state, i.e., $\mathbf{v}^{(i)}[t_j] = \lim_{\tau\to\infty}(\mathbf{f}[t_j](\tau)/\mathbf{f}[t_i](\tau))$. Hence, for any t_i, $\mathbf{f}_{ss} = \chi_i \cdot \mathbf{v}^{(i)}$, with χ_i the throughput of t_i. The vector of visit ratios is a right annuler of the incidence matrix $\mathbf{C}$, and therefore, in MTS systems, proportional to the unique T-semiflow. For MTS systems $\mathbf{f}_{ss} = \mathbf{f}_{ss}(\mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0})$, while $\mathbf{v}^{(i)} = \mathbf{v}_i(\mathcal{N})$, i.e., the visit ratio does not depend neither on $\boldsymbol{\lambda}$ nor on $\mathbf{m_0}$.

In the following Subsections some *at first glance* unexpected behaviors of continuous MTS systems are briefly shown.

5.1.1 Continuous is not an upper bound of discrete

It could be thought that, since continuisation removes some restrictions of the system, the throughput of the continuous system should be at least that of the discrete one. However, the throughput of a continuous PN is *not* in general an upper bound of the throughput of the discrete PN. For instance, in the net system in Figure 5.1(a), with $\boldsymbol{\lambda} = (3, 1, 1, 10)$, the throughput is 0.801 as discrete while it is only 0.535 as continuous. If the continuous marking is seen as a very large discrete marking, the reason for this "anomaly" lies in the non-monotonicity of the throughput under initial marking scaling (from $\mathbf{m_0}$ to $k \cdot \mathbf{m_0}, k > 0$) for discrete systems.

5.1.2 Non monotonicities

Like in discrete nets, the throughput of a continuous net system does not fulfill in general any monotonicity property, neither with respect to the *initial marking*, nor with respect to the *structure* of the net, nor with respect to the *firing rates*, $\boldsymbol{\lambda}$.

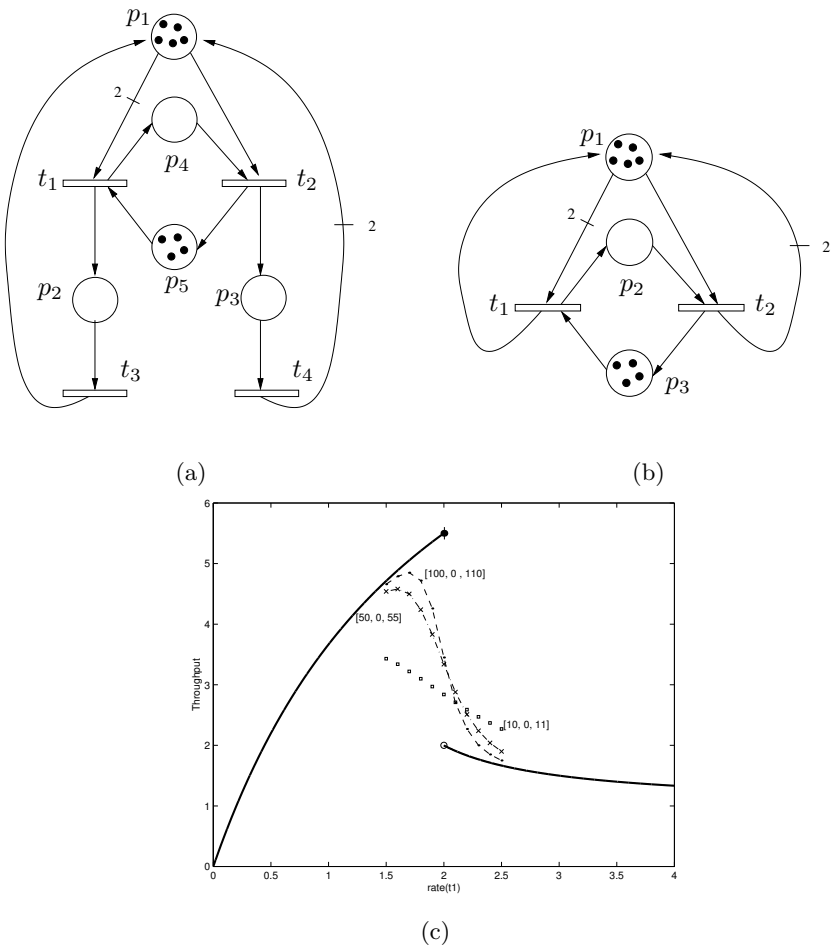


Figure 5.1: (a) A net system whose throughput as continuous is not an upper bound for the throughput as discrete, with $\lambda = (3, 1, 1, 10)$. (b), (c) For this net system, with $\boldsymbol{\lambda}[t_2] = 1$, increasing the rate of t_1 does not necessarily increase the throughput. Moreover a discontinuity appears at $\boldsymbol{\lambda}[t_1] = 2$.

For example, with respect to the initial marking, if in the timed net system of Figure 5.1(a) the marking of p_5 is augmented to 5, the systems deadlocks, i.e., the throughput goes down to 0. While if $\mathbf{m_0}[p_5]$ is reduced to 3 the throughput increases from 0.535 to 1.071. Notice that this token (i.e., resource) reduction is equivalent to adding a place "parallel" to p_5 (i.e., with an input arc from t_2 and an output arc to t_1), marked with 3 tokens. Hence, with respect to the net structure, adding constraints may increase the throughput!

Finally, an increase in a transition rate (for example, due to a replacement by a faster machine) may also lead to a decrease in the throughput. Moreover, a very small change may have a large effect. For example, the solid curve in Figure 5.1(c) represents how the throughput of the net system in Figure 5.1(b) changes if the rate of t_1 varies from 0 to 4, assuming $\boldsymbol{\lambda}[t_2] = 1$. Notice that even a *discontinuity* appears at $\boldsymbol{\lambda}[t_1] = 2$. Starting from the discrete underlying net system, this effect can be interpreted as the limit case when the number of discrete tokens goes to infinity (see the dotted lines in Figure 5.1(c) for the throughput of the discrete system with initial marking $\mathbf{m_0} = (10\ 0\ 11)$, $\mathbf{m_0} = (50\ 0\ 55)$ and $\mathbf{m_0} = (100\ 0\ 110)$).

5.2 Performance evaluation bounds

In this section it will be shown how bounds for the throughput of a continuous MTS system can be computed. Subsection 5.2.1 presents a non-linear programming problem that can be used to compute the *tight* bound of the system. It will be explained later what *tight* means here. A way to solve that programming problem consists in using a branch & bound algorithm (Subsection 5.2.2). Subsection 5.2.3 shows how the complexity of the algorithm can be reduced by pruning some nodes. In Subsection 5.2.4 the computation of lower bounds is addressed. In Subsection 5.2.5 the programming problem is *relaxed* leading to a linear programming problem (LPP), although this may lead to a non tight upper bound. A sufficient condition for the reachability of the bound computed by the obtained LPP is given in Subsection 5.2.5.

5.2.1 A non-linear programming problem for performance bounds

Let $\mathbf{m_{ss}}$ be the steady state marking of a continuous net system. For every $\tau > 0$, the following equations have to be verified:

$$\dot{\mathbf{m}}(\tau) = \mathbf{C} \cdot \mathbf{f}(\tau) \tag{5.1}$$

$$\mathbf{f}(\tau)[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\mathbf{m}(\tau)[p]}{\mathbf{Pre}[p,t]} \right\} \quad \forall\, t \in T \tag{5.2}$$

$$\mathbf{m}(0) = \mathbf{m_0} \tag{5.3}$$

$$\mathbf{m_{ss}} = \lim_{\tau \to \infty} \mathbf{m}(\tau) \tag{5.4}$$

The above equations can be relaxed as follows ($\boldsymbol{\mu}_{ss}$ and $\boldsymbol{\phi}_{ss}$ correspond to $\mathbf{m_{ss}}$ and $\mathbf{f}_{ss}$):

$$\boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}, \tag{5.5}$$

$$\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \right\} \quad \forall\, t \in T \tag{5.6}$$

$$\mathbf{C} \cdot \boldsymbol{\phi}_{ss} = \mathbf{0} \tag{5.7}$$

$$\boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \mathbf{0} \tag{5.8}$$

Equation (5.5) is obtained from (5.1) and (5.3), while (5.6) is a particularization of (5.2). Equation (5.6) may be seen as an application of the underlying idea in the Little's law for queueing systems (let $\theta[t] = \frac{1}{\boldsymbol{\lambda}[t]}$): $\theta[t] \cdot \boldsymbol{\phi}_{ss}[t] = \min_{p \in {}^\bullet t}\{\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\}$. For JF nets it can be rewritten as: $\boldsymbol{\phi}_{ss}[t] \cdot \mathbf{Pre}[{}^\bullet t, t] \cdot \theta[t] = \boldsymbol{\mu}_{ss}[{}^\bullet t]$, i.e., average flow of tokens by average residence time equals average number of tokens (this idea was first used in the field of Petri nets in [CAC+93]). Since $\mathbf{m_{ss}}$ is a steady state, from (5.1), $\mathbf{C} \cdot \mathbf{f}_{ss} = \mathbf{0}$ is deduced, and therefore (5.7) is immediately obtained.

This relaxation replaces the condition of being a reachable marking with being a solution of (5.5), the fundamental equation. That is, the information about the feasibility of the transient path is lost. Observe that the system is non-linear (*min* operator) and that it may have several solutions. For example, for the net system in Figure 5.2 with $\boldsymbol{\lambda} = (2, 1, 1)$, any marking $[10 - 5 \cdot \alpha, 4 \cdot \alpha - 3, \alpha, \alpha]$, with $1 \leq \alpha \leq 5/3$, verifies (5.5-5.8).

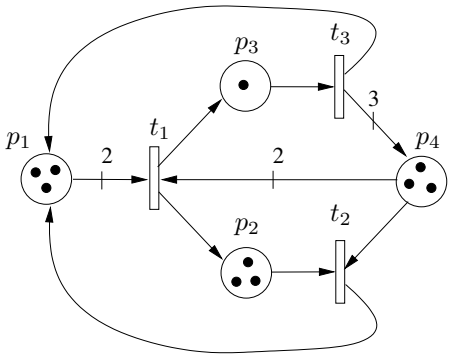


Figure 5.2: A continuous Petri net system.

Maximizing the flow of a transition (any of them, since all are related by the T-semiflow), an upper bound of the throughput in the steady state is obtained:

$$\begin{aligned}\max\{\boldsymbol{\phi}_{ss}[t_1] \mid \boldsymbol{\mu}_{ss} &= \mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma} \\ \boldsymbol{\phi}_{ss}[t] &= \boldsymbol{\lambda}[t]\cdot\min_{p\in{}^\bullet t}\left\{\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\right\}\ \forall t\in T \\ \mathbf{C}\cdot\boldsymbol{\phi}_{ss} &= \mathbf{0} \\ \boldsymbol{\mu}_{ss},\boldsymbol{\sigma} &\geq \mathbf{0}\}\end{aligned} \tag{5.9}$$

If the flow in (5.9) is *minimized* instead of maximized (see Subsection 5.2.4) a very similar algorithm can be used for the computation of lower bounds.

Let us consider the following Proposition that will help to understand better the kind of solutions obtained in (5.9).

Proposition 5.1. [STC98] If $\mathcal{N}$ is consistent and conservative, the following statements are equivalent:

1. $\boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}\geq\mathbf{0}$
2. $\forall\ \mathbf{y}\geq\mathbf{0}$ such that $\mathbf{y}\cdot\mathbf{C}=\mathbf{0}$, then $\mathbf{y}\cdot\boldsymbol{\mu}_{ss}=\mathbf{y}\cdot\mathbf{m_0}$

This means that relaxing the conditions on $\mathbf{m_{ss}}$ to being a solution of the fundamental equation (that is, making the system insensitive to the transient), is equivalent to saying that the solution is insensitive to the initial marking distribution inside the P-semiflows. More precisely, it only depends on the loads of the P-semiflows, $\mathbf{y}\cdot\mathbf{m}=\mathbf{y}\cdot\mathbf{m_0}$.

Notice that the solution of (5.9) is always "reachable" in the sense that with a suitable initial distribution of the tokens inside the P-semiflows (for instance, with the same steady state distribution), its associated throughput can be obtained. This is why, it will be said that the solution is a *tight* bound.

Nevertheless, the non-linear programming problem in (5.9) is difficult to solve due to the "min" condition coming from equation (5.6). When a transition t has a single input place, equation (5.6) reduces to equation (5.10). When t has more than one input place, then equation (5.6) can be relaxed (linearized) as equation (5.11).

$$\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \text{if} \quad p={}^\bullet t \tag{5.10}$$

$$\boldsymbol{\phi}_{ss}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \forall\ p\in{}^\bullet t\ \text{ otherwise} \tag{5.11}$$

This way, one has a single linear programming problem (LPP) (5.12) defined by equalities (5.5), (5.7) and (5.10), and inequalities (5.8) and (5.11), that can be solved in *polynomial time*.

$$
\begin{aligned}
\max\{\boldsymbol{\phi}_{ss}[t_1] \mid\ & \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma} \\
& \boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \text{if} \quad p = {}^\bullet t \\
& \boldsymbol{\phi}_{ss}[t] \leq \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^\bullet t \ \text{ otherwise} \\
& \mathbf{C}\cdot\boldsymbol{\phi}_{ss} = \mathbf{0} \\
& \boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \mathbf{0}\}
\end{aligned}
\tag{5.12}
$$

Unfortunately, this LPP provides in general a non tight bound, i.e, the solution may be non reachable for any distribution of the tokens verifying the P-semiflow load conditions, $\mathbf{y}\cdot\mathbf{m_0}$. This occurs when none of the input places of a transition really restricts the flow of that transition. When this happens, the marking does not define the steady state (the flow of that transition would be larger).

For example, for the net system in Figure 5.2 with $\boldsymbol{\lambda} = \mathbf{1}$, the optimum of the LPP is $\mathbf{f}_{ss}[t_1] = 1.25$. This value is obtained for $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 3.25$, $\mathbf{m}[p_3] = 1.25$, and $\mathbf{m}[p_4] = 2.5$. Under this marking the throughput of t_2 would be 2.5, while for the rest of the transitions, it is 1.25. Since $\mathbf{v}^{(1)} = \mathbf{1}$, this cannot be the steady state. It can be seen that this happens for any maximal solution of this particular LPP. Hence the LPP in this case provides a non-reachable bound of the throughput. In fact, the maximum throughput for this system is 0.75.

5.2.2 Towards a Branch & Bound (B & B) algorithm

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 in (5.9). The problem is that there is no way to know in advance which of the input places should restrict the flow. A B & B algorithm can be used to compute a steady state marking that fulfills what (5.12) expresses. If the marking solution of (5.12) does not correspond to a steady state (i.e., there is at least one transition such that all its input places have "more than the necessary" tokens) choose one of the synchronizations and solve the set of LPPs that appear when each one of the input places are assumed to be defining the flow. That is, build a set of LPPs by adding an equation that relates the marking of each input place with the flow of the transition. These subproblems become children of the root search node. The algorithm is applied recursively, generating a tree of subproblems. If an optimal steady state marking is found to a subproblem, it is a feasible steady state marking, but not necessarily globally optimal. Since it is feasible, it can be used to prune the rest of the tree: if the solution of the LPP for a node is smaller than the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented

by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned.

The recursive Algorithm 5.2 sketches how the B & B algorithm works. The inputs of the algorithm are the net, the initial marking and the set of pairs (p, t) such that the marking of p is wanted to define the flow of t in the steady state. This set of pairs is denoted by *eqs* (because it represents the equalities, i.e., the constraints for the firing of transitions) and in the first call to the algorithm will be equal to those (p, t) such that $p = {}^\bullet t$. Successive calls to the algorithm will increase the set *eqs* in order to force the flow of the rest of transitions to be defined by the marking of an input place. The output of the algorithm is given by the global variable *bound*. The procedure *max_LPP* solves the LPP(5.12), i.e., equations (5.5), (5.7), (5.10) applied on the pairs (p, t) in *eqs*, inequality (5.11) applied on every input place of the transitions that are not in *eqs*, and inequality (5.8). Let us assume that *max_LPP* returns a scalar x corresponding to the solution of the LPP, and a set nt (non-satisfied transitions) containing those transitions for which their flow according to x and the vector of visit ratios is less than it should be according to the markings of their input places.

Algorithm 5.2 (Branch & Bound, upper bounds).

Global Variable: *bound* := 0 % Initially equal to 0
Input: ($\mathcal{N}$, $\mathbf{m_0}$, eqs)
Output: *bound*
Begin
  $(x, nt) := max\_LPP(\mathcal{N}, \mathbf{m_0}, eqs)$
  If $x \leq$ *bound* or the LPP was infeasible **then**
    % Do nothing. This node is pruned.
  Else
    If $nt = \emptyset$ **then** % The solution represents a steady state
      $bound := max(bound, x)$
    Else
      take a $t \in nt$ do
      For every $p \in {}^\bullet t$ do
        $eqs := eqs \cup (p, t)$
        Branch_Bound($\mathcal{N}$, $\mathbf{m_0}$, eqs)
      End_For
    End_If
  End_If
End

The system model in Figure 5.3 is a MTS system (but not a marked graph because of the Mi_Idle places). It represents a flexible manufacturing system composed of three machines: M1, M2 and M3. Parts of type A are processed first in machine M1 and

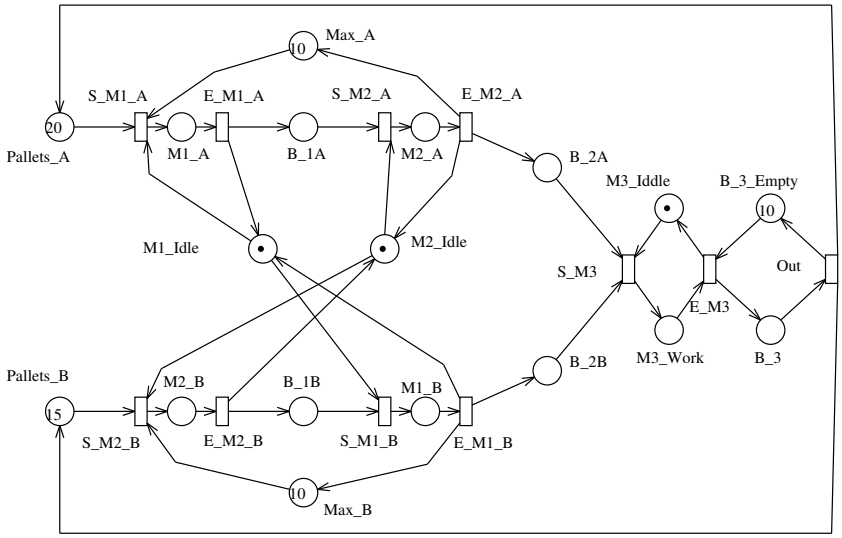


Figure 5.3: A PN model of a flexible manufacturing system.

then in machine M2, while parts of type B are processed first in M2 and then in M1. The intermediate products are stored in buffer B_1A and B_1B, and the final parts in buffers B_2A and B_2B, respectively. Machine M3 takes a part A and a part B and assembles the final product, that is stored in B_3 until its removal. In B_3 there is space at most for 10 products. There can be at most 10 parts of type A and 10 parts of type B either in B_1A and B_1B, or being processed by M1 and M2. Parts are moved in pallets all along the process, and there are 20 pallets of type A and 15 pallets of type B. The firing speeds of transitions are: $\boldsymbol{\lambda}$[Out] = $\boldsymbol{\lambda}$[S_M1_A] = $\boldsymbol{\lambda}$[S_M2_A] = $\boldsymbol{\lambda}$[S_M1_B] = $\boldsymbol{\lambda}$[S_M2_B] = $\boldsymbol{\lambda}$[S_M3] = 1 , $\boldsymbol{\lambda}$[E_M3] = $\boldsymbol{\lambda}$[E_M2_A] = 1/4, $\boldsymbol{\lambda}$[E_M1_A] = $\boldsymbol{\lambda}$[E_M2_B] = 1/3, $\boldsymbol{\lambda}$[E_M1_B] = 1/5.

The visit ratio of the system is $\mathbf{v}^{(1)} = \mathbf{1}$, that is, in the steady state all the operations have to be executed at the same rate (this is imposed by the assembly of one part A and one part B). A solution of the original LPP (5.12) is $\mathbf{f}_{ss}$[Out] = 0.111, $\mathbf{m}$[Pallets_A] = 9, $\mathbf{m}$[B_2A] = $\mathbf{m}$[Max_A] = $\mathbf{m}$[B_2B] = $\mathbf{m}$[Max_B] = $\mathbf{m}$[M3_Idle] = $\mathbf{m}$[B_3] = $\mathbf{m}$[M1_Idle] = 0.111, $\mathbf{m}$[M1_A] = 0.333, $\mathbf{m}$[B_1A] = 9.111, $\mathbf{m}$[Pallets_B] = 4, $\mathbf{m}$[M2_A] = 0.444, $\mathbf{m}$[M2_B] = 0.333, $\mathbf{m}$[B_1B] = 9, $\mathbf{m}$[M1_B] = 0.555, $\mathbf{m}$[M2_Idle] = 0.222, $\mathbf{m}$[M3_Work] = 0.888, $\mathbf{m}$[B_3_Empty] = 9.888. This solution corresponds to the root node, see Node 1 in Figure 5.4, of the tree that the B & B algorithm computes when applied to the flexible manufacturing system.

According to the value obtained by the LPP, $\mathbf{f}_{ss}$[Out] = 0.111, and the vector of

visit ratios, the throughput of all the transitions in the steady state should be 0.111. However, observe that if one considers the markings obtained for the input places of transition S_M2_A and E_M3, the throughput is greater than 0.111 (in the first call to the *branch_bound* algorithm nt equals $\{S\_M2\_A, E\_M3\}$). That is, the obtained marking is not a steady state marking. If one first focuses on S_M2_A, two LPPs should be built since it has two input places, adding in each one an equation for S_M2_A. If one forces the throughput to be defined by M2_Idle (i.e., $\boldsymbol{\phi}_{ss}[S\_M2\_A] = \boldsymbol{\lambda}[S\_M2\_A] \cdot \frac{\boldsymbol{\mu}_{ss}[M2\_Idle]}{\mathbf{Pre}[M2\_Idle, S\_M2\_A]}$), Node 2, the system is *infeasible*, while if one add a restriction for B_1A, Node 3, the solution is the same but for $\mathbf{m}$[Pallets_A] = 18, $\mathbf{m}$[B_1A] = 0.111, and $\mathbf{m}$[Max_A] = 9.111. Now, the only problem is E_M3. If one adds an equation for B_3_Empty, Node 4, the system is *infeasible*. Adding an equation for M3_Work, Node 5, modifies $\mathbf{m}$[Pallets_A] = 18.444, $\mathbf{m}$[Pallets_B] = 4.444, $\mathbf{m}$[M3_Idle] = 0.555, and $\mathbf{m}$[M3_Work] = 0.444. This is a steady state marking and the throughput associated to it is equal to the one obtained with the original LPP, $\mathbf{f}_{ss}$[Out] = 0.111. No higher throughput may exist and no more branching is needed.

5.2.3 Pruning nodes in the B & B algorithm

The branching process developed in the algorithm is based on associating transitions with input places, i.e., forcing the throughput of a transition to be defined by one of its input places. The number of nodes of the tree in the worst case is $1 + \sum_{t_i \in T, |{}^\bullet t_i| > 1} \Pi^i_{j=1, |{}^\bullet t_j| > 1} |{}^\bullet t_j|$ where the transitions are sorted according to their number of input places ($|{}^\bullet t_1| \geq |{}^\bullet t_2| \geq |{}^\bullet t_3| ...$). Number 1 stands for the root node and each element of the sum stands for one transition (one level of the tree) with several input places. In the worst case the algorithm has to explore the complete tree and solve a linear programming problem, whose complexity is polynomial, per node. However, in most real cases such exhaustive exploration is not required since some branches can be pruned according to the B & B algorithm. Moreover, some considerations will be done in this Subsection that allow one to further reduce the number of nodes to be explored.

Places can be seen as suppliers of fluid, i.e., clients, to their output transitions. Transitions can be seen as stations demanding fluid to their input places. Since MTS systems are being considered, in the steady state, the throughput of the transitions has to be proportional to the vector of visit ratios. Clearly, in the steady state, not all the output transitions of a given place, p, are equally *fluid-demanding*, i.e., one output transition t_1 may require a higher marking of p than other output transition t_2, in order to fire according to its visit ratio. In the steady state, a place p can at most define the throughput of that output transition that is demanding the greatest amount of fluid to p. Thus, the B & B algorithm should avoid the exploration of those nodes that associate (the marking of) a place to (the flow of) an output transition

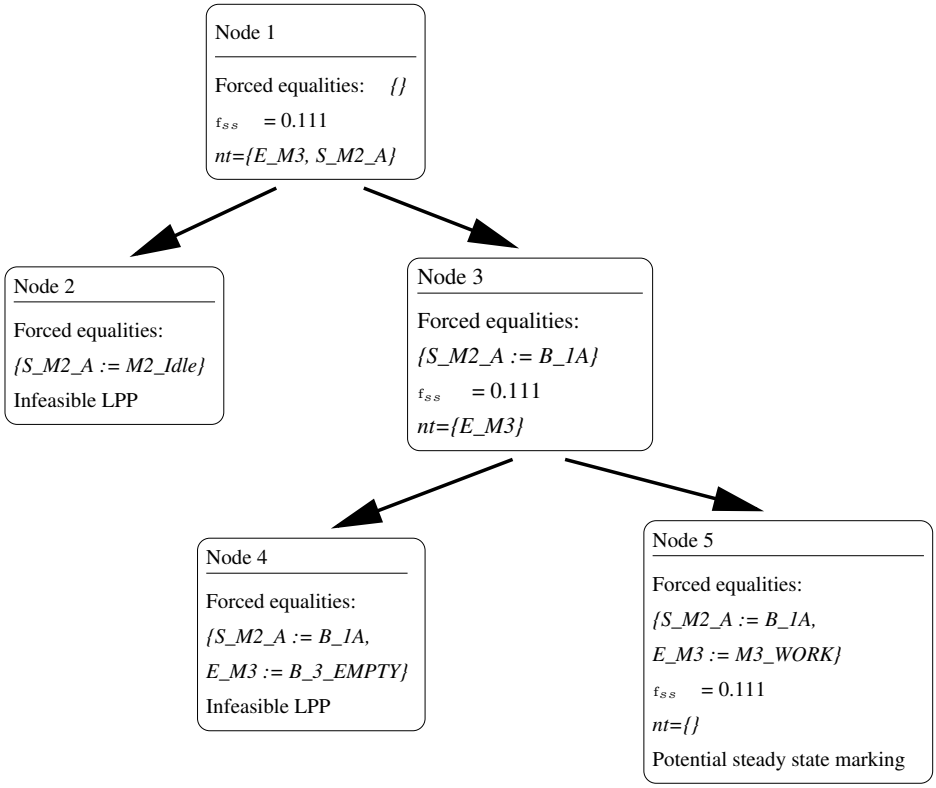


Figure 5.4: Tree obtained by the B & B algorithm for the computation of upper bounds applied on the system in Figure 5.3.

that is not its most fluid-demanding transition.

Let us reconsider Equation (5.6) in order to compute the most fluid-demanding transition for a given place p in the steady state. A simple relaxation of (5.6) consists in just looking if each place has enough fluid to fire all its output transitions:

$$\boldsymbol{\mu}_{ss}[p] \geq \max_{t \in p^{\bullet}} \left\{ \frac{\mathbf{Pre}[p,t] \cdot \boldsymbol{\phi}_{ss}[t]}{\boldsymbol{\lambda}[t]} \right\} \tag{5.13}$$

The right part of Equation (5.13) can be seen as the amount of fluid demanded to place p in the steady state by each of its output transitions. The transition giving the maximum is the most fluid-demanding transition, and so, it is the only transition that can be associated to place p in the B & B algorithm. See Subsection 4.3.2 for details on how to compute the most fluid-demanding transition of a given place.

Let us consider the system in Figure 5.2 with $\boldsymbol{\lambda} = \mathbf{1}$ to show how a node can be

pruned by using the above reasonings. In principle, in the steady state the throughput of transition t_2 could be defined by any of its input places p_2 or p_4. Notice that p_4 has two output transitions t_1 and t_2, and so in the steady state it has to supply enough fluid for both transitions. The vector of visit ratios of this system is $\mathbf{v}^{(1)} = \mathbf{1}$. Therefore, in the steady state the throughput of every transition is the same. Thus, since $\boldsymbol{\lambda} = \mathbf{1}$ also the enabling degree of every transition should be the same in the steady state. Let us assume that in the steady state the enabling degree of t_2 is given by p_4. This implies that the enabling degree of t_1 is at most half the enabling degree of t_2 since there is an arc with weight 2 going from p_4 to t_2. In other words, t_1 is the most fluid-demanding transition of p_4: t_1 is demanding the double of fluid to p_4 than t_2. Hence, there cannot exist a non dead steady state marking at which p_4 is defining the flow of t_2.

Recall that for the system in Figure 5.2 the LPP (5.12) yields $\mathbf{f}_{ss}[t_1] = 1.25$ with $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 3.25$, $\mathbf{m}[p_3] = 1.25$, and $\mathbf{m}[p_4] = 2.5$. Under this solution the throughput of t_2 would not be 1.25 and therefore it cannot represent a steady state marking. If one tries to force the throughput of t_2 to be defined by p_4, an infeasible LPP will be obtained. Therefore, the B & B algorithm must avoid the computation of the node that associates p_4 to t_2. Forcing the throughput of transition t_2 to be defined by p_2 the solution of the LPP is $\mathbf{f}_{ss}[t_1] = 0.75$ with $\mathbf{m}[p_1] = 5.5$, $\mathbf{m}[p_2] = 0.75$, $\mathbf{m}[p_3] = 0.75$, and $\mathbf{m}[p_4] = 1.5$. That is a steady state marking, and therefore 0.75 is an upper bound for the throughput of the system.

Taking into account the most fluid-demanding transitions of places the complexity of the B & B algorithm presented in the previous Subsection is reduced in many cases. Since a place will be associated only to its most fluid-demanding transition, the higher the number of output transitions of places, the higher the reduction of the complexity of the algorithm. Unfortunately, if all the places have a single output transition the complexity of the algorithm is not reduced. Anyway, the computation of the most fluid-demanding transitions requires only the structure of the timed net, and so it can be easily added to the B & B algorithm.

5.2.4 Lower bounds and exact throughput

Lower throughput bounds can be computed in a very similar way to upper bounds by means of a B & B algorithm. In this case, the goal function has to be minimized instead of maximized.

Let us compute the upper and lower bounds for the MTS system in Figure 5.5 with initial marking $\mathbf{m_0} = (4\ 0\ 1)$ and $\boldsymbol{\lambda} = (1\ 2)$. The visit ratio for the underlying net is $\mathbf{v}^{(1)} = (1\ 1)$. For the upper bound, the application of the B & B algorithm discussed in the previous section yields $\mathbf{f}_{ss}[t_1] = 2.5$ with $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 1.5$ and $\mathbf{m}[p_3] = 2.5$ for the initial LPP (without any branching). That is a steady state marking, and so, a suitable upper bound has been computed.

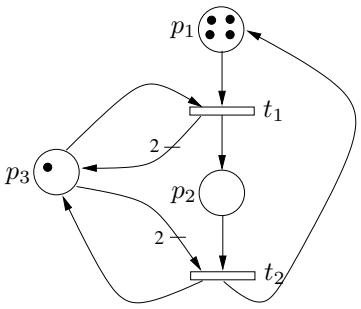


Figure 5.5: A PN system with different upper and lower throughput bounds.

Minimizing the throughput of that LPP, one obtains $\mathbf{f}_{ss}[t_1] = 0$ with $\mathbf{m}[p_1] = 1.45$, $\mathbf{m}[p_2] = 2.55$ and $\mathbf{m}[p_3] = 3.55$. At this marking, neither t_1 nor t_2 have a throughput of 0 as established by the solution of the LPP. This is a non surprising result since every transition has more than one input place and so $\mathbf{0}$ is a trivial solution of the equations in (5.12). Figure 5.6 represents the tree explored by the B & B algorithm to compute the lower throughput bound for the system in Figure 5.5.

If the throughput of transition t_1 is forced to be defined by the marking of p_3, Node 2, the LPP yields $\mathbf{f}_{ss}[t_1] = 2$ with $\mathbf{m}[p_1] = 3$, $\mathbf{m}[p_2] = 1$ and $\mathbf{m}[p_3] = 2$ which is a steady state marking. Hence no more branching is required from this node. If transition t_1 is controlled by p_1, Node 3, one obtains $\mathbf{f}_{ss}[t_1] = 0$ with $\mathbf{m}[p_1] = 0$, $\mathbf{m}[p_2] = 4$ and $\mathbf{m}[p_3] = 5$. In this case the throughput of t_2 is not defined by any of the markings of its input places. From this node it is possible to associate t_2 to either p_2 or p_3. If t_2 is associated to p_2, Node 5, an infeasible LPP is obtained. If t_2 is associated to p_3, Node 4, the LPP gives $\mathbf{f}_{ss}[t_1] = 2.5$ with $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 1.5$ and $\mathbf{m}[p_3] = 2.5$, which is a steady state marking. At this point, the B & B algorithm finishes and it can be concluded that $\mathbf{f}_{ss}[t_1] = 2$ is a suitable lower bound.

Considering again the system in Figure 5.2, the lower bound obtained after the application of the B & B algorithm is $\mathbf{f}_{ss}[t_1] = 0.75$ with $\mathbf{m}[p_1] = 5.5$, $\mathbf{m}[p_2] = 0.75$, $\mathbf{m}[p_3] = 0.75$, and $\mathbf{m}[p_4] = 1.5$. This lower bound is identical to the upper bound obtained for this system in Subsection 5.2.3. This means that the exact throughput of the system in the steady state has just been computed. The B & B algorithm focuses on the initial load of the P-semiflows and not on the initial marking of each place, therefore, it holds that for any initial distribution of the given load, the system in Figure 5.2 will reach a steady state in which $\mathbf{f}_{ss}[t_1] = 0.75$.

In Subsection 5.2.2 the application of the B & B algorithm on the system in Figure 5.3 yielded $\mathbf{f}_{ss}[\text{Out}] = 0.111$ as an upper bound for the throughput. If the B & B is applied to compute the lower bound for that system the same value is obtained. That is, $\mathbf{f}_{ss}[\text{Out}] = 0.111$ is the exact throughput of the system in the steady state.

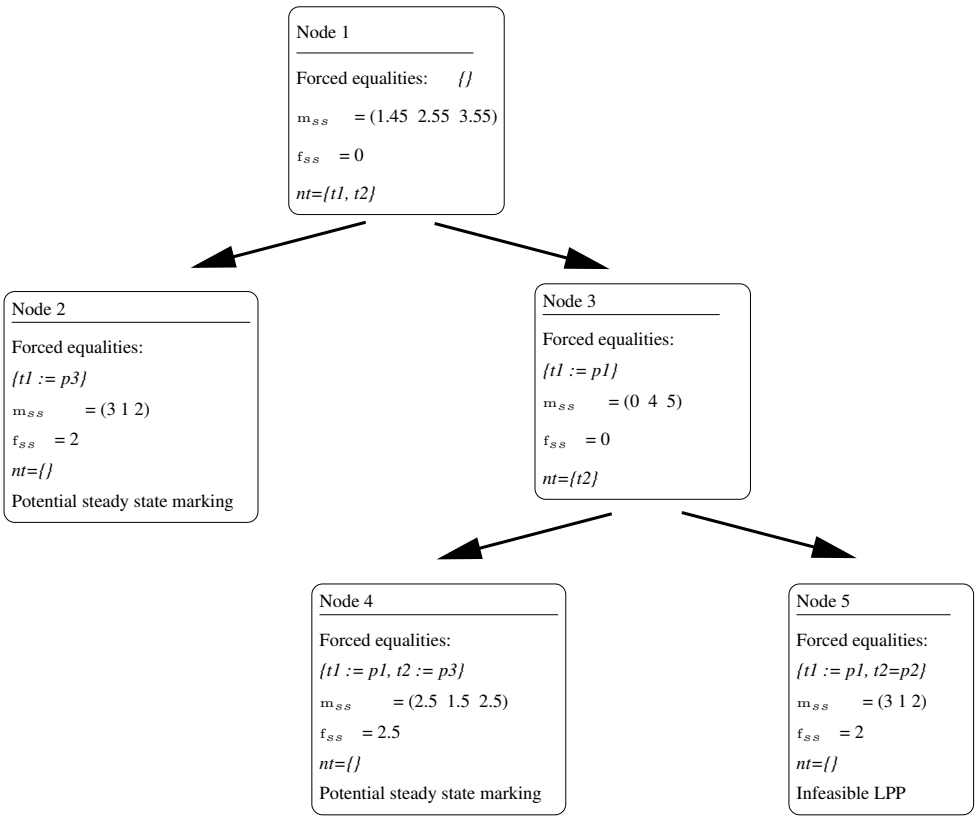


Figure 5.6: Tree obtained by the B & B algorithm for the computation of lower bounds applied on the system in Figure 5.5.

If the system is considered as a discrete Petri net the number of reachable states is 1357486. The throughput of the discrete system computed by solving the associated Markov chain is $\mathbf{f}_{ss}[\text{Out}] = 0.1090$.

5.2.5 Branching elimination for the computation of upper bounds

Let us consider again the problem defined by the LPP in (5.12). As it has been done in Subsection 5.2.3, Equation (5.6) can be relaxed to (5.13). The following single LPP can be obtained to compute an upper throughput bound:

$$\begin{aligned}\max\{\boldsymbol{\phi}_{ss}[t_1] \mid\ & \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma} \\ & \boldsymbol{\mu}_{ss}[p] \geq \max_{t\in p^\bullet}\left\{\frac{\mathbf{Pre}[p,t]\cdot\boldsymbol{\phi}_{ss}[t]}{\boldsymbol{\lambda}[t]}\right\}\ \forall p\in P \\ & \mathbf{C}\cdot\boldsymbol{\phi}_{ss} = \mathbf{0} \\ & \boldsymbol{\sigma},\boldsymbol{\mu}_{ss}\geq\mathbf{0}\}\end{aligned} \tag{5.14}$$

Since in MTS systems $\mathbf{v}^{(1)}$ is completely defined, if $\boldsymbol{\phi}_{ss} = \chi\cdot\mathbf{v}^{(1)}$, LPP (5.14) can be written as:

$$\begin{aligned}\max\{\chi \mid\ & \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma} \\ & \boldsymbol{\mu}_{ss} \geq \chi\cdot\mathbf{PD},\quad \boldsymbol{\sigma},\boldsymbol{\mu}_{ss}\geq\mathbf{0}\}\end{aligned} \tag{5.15}$$

where $\mathbf{PD}[p] = \max_{t\in p^\bullet}\left\{\frac{\mathbf{Pre}[p,t]\cdot\mathbf{v}^{(1)}[t]}{\boldsymbol{\lambda}[t]}\right\}$.

Defining $\alpha = 1/\chi$ and $\boldsymbol{\sigma}' = 1/\chi\cdot\boldsymbol{\sigma}$, (5.15) reduces to:

$$\min\{\alpha \mid \alpha\cdot\mathbf{m_0} + \mathbf{C}\cdot\boldsymbol{\sigma}'\geq\mathbf{PD},\quad \boldsymbol{\sigma}'\geq\mathbf{0}\} \tag{5.16}$$

The dual of this LPP is:

$$\max\{\mathbf{y}\cdot\mathbf{PD} \mid \mathbf{y}\cdot\mathbf{C}\leq\mathbf{0},\ \mathbf{y}\cdot\mathbf{m_0}\leq 1,\ \mathbf{y}\geq\mathbf{0}\} \tag{5.17}$$

One of the formulations of the *alternatives theorem* [Mur83] states that the following two statements are equivalent:

$$\begin{aligned}&1.\ \exists\ \mathbf{x}>\mathbf{0} \text{ such that } \mathbf{C}\cdot\mathbf{x}\geq\mathbf{0} \\ &2.\ \forall\ \mathbf{y}\geq\mathbf{0} \text{ such that } \mathbf{y}\cdot\mathbf{C}\leq\mathbf{0} \text{ then } \mathbf{y}\cdot\mathbf{C}=\mathbf{0}\end{aligned} \tag{5.18}$$

Since MTS nets are consistent, the first statement of (5.18) is true, and therefore the second one is also true. Hence the condition $\mathbf{y}\cdot\mathbf{C}\leq\mathbf{0}, \mathbf{y}\geq\mathbf{0}$ in (5.17) can be replaced by $\mathbf{y}\cdot\mathbf{C}=\mathbf{0}, \mathbf{y}\geq\mathbf{0}$. Moreover, since $\mathbf{y}\cdot\mathbf{PD}$ is being maximized, the solution must verify $\mathbf{y}\cdot\mathbf{m_0} = 1$ (otherwise a better result can be obtained with $\beta\cdot\mathbf{y},\ \beta = 1/(\mathbf{y}\cdot\mathbf{m_0})$).

Proposition 5.3. Let γ be the solution of:

$$\begin{aligned}\gamma = \max\{\mathbf{y}\cdot\mathbf{PD} \mid\ & \mathbf{y}\cdot\mathbf{C} = \mathbf{0} \\ & \mathbf{y}\cdot\mathbf{m_0} = 1,\ \mathbf{y}\geq\mathbf{0}\}\end{aligned} \tag{5.19}$$

The throughput in the steady state verifies $\mathbf{f}_{ss}\leq\frac{1}{\gamma}\mathbf{v}^{(1)}$

Intuitively, the idea of (5.19) is to find the slowest subnet among those generated by the elementary P-semiflows. In other words, the bound is obtained by looking at the bottleneck P-semiflow. The complexity of finding the bottleneck P-semiflow is polynomial since it is obtained by solving a LPP. In [CS92] a similar result was obtained for discrete systems in which conflicts were forbidden except among immediate transitions.

Reachability of the bound

For each marking $\mathbf{m}$, its T-coverture (T-cov($\mathbf{m}$)) is defined as the set of places that restrict the flow of the transitions.

Definition 5.4. Given a net system, the *T-coverture* at a marking $\mathbf{m}$, is

$$\text{T-cov}(\mathbf{m}) = \{p \mid \exists t \in p^\bullet \text{such that } \mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \mathbf{m}[p]/\mathbf{Pre}[p,t]\}$$

The T-coverture of a system at a marking $\mathbf{m}$ can be also defined as the set of places contained in PT-set($\mathbf{m}$) (see Subsection 1.3.2).

A characterization can be obtained for the solution of (5.19) being the exact value. Given a vector $\mathbf{v}$, let us denote as $\|\mathbf{v}\|$ its support, i.e., the set of its non-zero components.

Proposition 5.5. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a MTS continuous system.

The flow computed with (5.19) (or (5.15)) is the flow in the steady state iff the T-coverture at the steady state, T-cov($\mathbf{m_{ss}}$), contains the support of a P-semiflow.

Moreover, the maximum of (5.19) is reached for the P-semiflow contained in the T-coverture.

Proof. Let $\mathbf{m_{ss}}$ be the steady state marking, $\mathbf{f}_{ss} = \chi_1 \cdot \mathbf{v}^{(1)}$ the flow vector associated to this state, and γ the solution of (5.19). Applying (5.19), $\chi_1 \leq 1/\gamma$.

For "$\Rightarrow$", assume that $\mathbf{y}_0$ is a P-semiflow such that the maximum of the LPP is reached. If its support is not contained in T-cov($\mathbf{m_{ss}}$), a place $p \in \|\mathbf{y}_0\|$ exists such that $\mathbf{m_{ss}}[p] > \max_{t \in p^\bullet}\{\mathbf{Pre}[p,t] \cdot \chi_1 \cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\} = \chi_1 \cdot \mathbf{PD}[p]$. Hence, $\mathbf{y}_0 \cdot \mathbf{m_{ss}} > \chi_1 \cdot \mathbf{y}_0 \cdot \mathbf{PD}$, and $1/\chi_1 > \mathbf{y}_0 \cdot \mathbf{PD} = \gamma$, contradiction.

For "$\Leftarrow$", let $\mathbf{y}_0$ be a P-semiflow such that $\|\mathbf{y}_0\| \subseteq$ T-cov($\mathbf{m_{ss}}$) and $\mathbf{y}_0 \cdot \mathbf{m_0} = 1$. Then, for every $p \in \|\mathbf{y}_0\|$, a transition $t \in p^\bullet$ exists such that $\mathbf{m_{ss}}[p] = \mathbf{Pre}[p,t] \cdot \chi_1 \cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]$. Hence, $\mathbf{m_{ss}}[p] = \chi_1 \cdot \max_{t \in p^\bullet}\{\mathbf{Pre}[p,t] \cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\} = \chi_1 \cdot \mathbf{PD}[p]$.

Therefore $\gamma \geq \mathbf{y}_0 \cdot \mathbf{PD} = \mathbf{y}_0 \cdot \mathbf{m_{ss}}/\chi_1 = 1/\chi_1$. Then $1/\chi_1 = \gamma$. □

From Prop. 5.5 the following Corollary is obtained:

Corollary 5.6. Let $\mathcal{N}$ be a MTS continuous net. If the P-subnet defined by any T-coverture contains a P-semiflow, then the flow at the steady state can be computed in polynomial time with the LPP (5.19).

5.3 Extending the subclass of nets: MTS reducible nets

One interesting property of MTS nets is that the vector of visit ratios only depends on the net structure, i.e., $\mathbf{v}^{(1)} = \mathbf{v}^{(1)}(\mathcal{N})$. Therefore, as it has been seen, knowing the flow in the steady state of one transition, the flow of the rest of transitions is trivially computed. However, the subclass of nets for which the vector of visit ratios does not depend on the initial marking is larger than the class of MTS nets. If one considers the net in Figure 5.7(a) with $\boldsymbol{\lambda} = (1\ 1\ 1\ 1)$, one will realize that the flow through transitions in the steady state is always proportional to the vector (2 1 2 1), that is the flow through transitions t_2 and t_4 is double than the flow through transitions t_1 and t_3, independently of the initial marking. The reason for this fact is that given a continuous net (not necessarily MTS), the following is verified:

$$\frac{\mathbf{f}[t_i]}{\mathbf{Pre}[p,t_i]\cdot\boldsymbol{\lambda}[t_i]} = \frac{\mathbf{f}[t_j]}{\mathbf{Pre}[p,t_j]\cdot\boldsymbol{\lambda}[t_j]}\ \forall\ t_i,t_j \text{ in CEQ relation}, \forall\, p \in {}^\bullet t_i \tag{5.20}$$

Where CEQ stands for continuous equal conflict (see Section 1.2). In this section, the results obtained in Section 5.2 will be extended to a larger class of nets, the class of *mono-T-semiflow reducible* nets (MTSR), for which $\mathbf{v}^{(1)} = v^{(1)}(\mathcal{N}, \boldsymbol{\lambda})$, i.e., $\mathbf{v}^{(1)}$ does not depend on the initial marking.

Definition 5.7. Let $\mathcal{N}$ be a consistent and conservative PN and $\boldsymbol{\lambda}$ a speeds vector. It will be said that $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is *mono T-semiflow reducible* (MTSR) if the following system has a unique solution:

$$\begin{aligned} &\mathbf{C}\cdot\mathbf{v}^{(1)} = \mathbf{0} \\ &\frac{\mathbf{v}^{(1)}[t_i]}{\mathbf{Pre}[p,t_i]\cdot\boldsymbol{\lambda}[t_i]} = \frac{\mathbf{v}^{(1)}[t_j]}{\mathbf{Pre}[p,t_j]\cdot\boldsymbol{\lambda}[t_j]}\ \forall\, t_i,t_j \text{ in CEQ relation}, \forall\, p \in {}^\bullet t_i \\ &\mathbf{v}^{(1)}[t_1] = 1 \end{aligned} \tag{5.21}$$

Every continuous MTSR net can be *reduced* to an "equivalent" MTS net with identical behavior. The *reduction rule* consists in merging those transitions in CEQ relation into only one flow-equivalent transition. The arcs and the firing speed, $\boldsymbol{\lambda}$, of the equivalent transition have to be such that they preserve the evolution of their input and output places. This can be achieved with simple arithmetic operations on the weights of the input/output arcs and the firing speeds of the original transitions. Figure 5.8 sketches how two transitions in CEQ relation can be merged to a single one. It can be checked that the evolution of the input places is preserved and so is the flow associated to the output arcs. An iterative merger on every couple of transitions

in CEQ relation leads to a net without CEQ's and to a MTS if the original net was MTSR. Notice that the arc weights of the resulting net may not be natural numbers. Nevertheless, this is not a problem for any of the properties being considered.

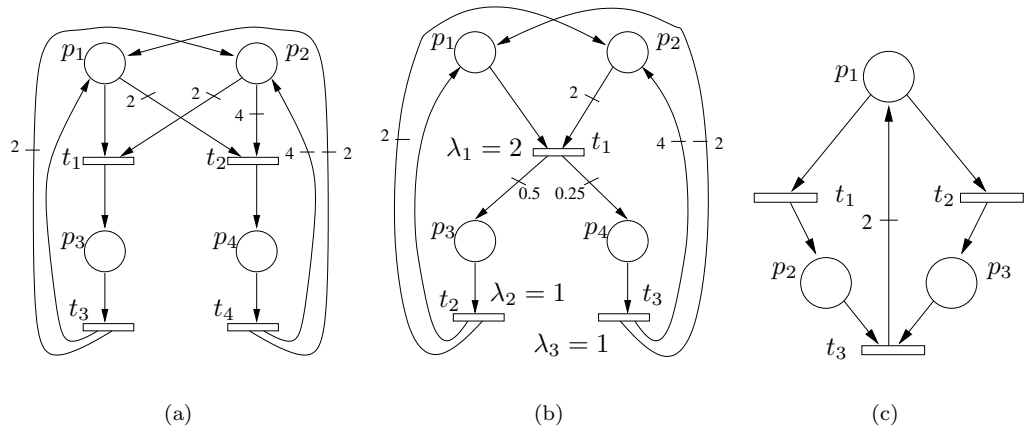
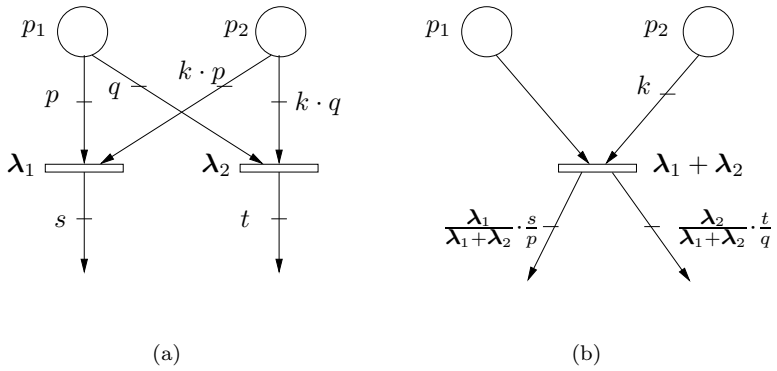


Figure 5.7: (a) A MTSR (but not MTS net) for every $\boldsymbol{\lambda} > \mathbf{0}$. (b) Equivalent MTS net given $\boldsymbol{\lambda} = \mathbf{1}$. (c) A MTS net that belongs to MTSR iff $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2$.

Observe that if the input and output arc weights of a transition are multiplied by a constant, the evolution of the input and output places is the same. However, the flow through the transition varies in an inverse proportion to the constant. Therefore, by varying that constant, it is possible to reduce a MTSR net to an infinite number of equivalent MTS nets.

The net in Figure 5.7(a) is MTSR (for $\boldsymbol{\lambda} = \mathbf{1}$ its equivalent MTS net is depicted in Figure 5.7(b)) but not MTS since it has two T-semiflows. The net in Figure 5.7(c) with $\boldsymbol{\lambda} = (1\ 1\ 1)$ belongs to both MTS and MTSR. Nevertheless, it should be noticed that the class MTSR does not include the class MTS: The net in Figure 5.7(c) with $\boldsymbol{\lambda} = (1\ 2\ 1)$ belongs to MTS but not to MTSR. However, disregarding those nets that belong only to MTS does not imply a loss of generality since their steady state throughput is zero, that is, they are not *structurally timed-live*. Therefore, it has no sense computing throughput bounds for systems like the one in Figure 5.7(c) with $\boldsymbol{\lambda} = (1\ 2\ 1)$, because it is null. The diagram in Figure 5.9 shows the relationship between the classes MTS and MTSR.

Extending the results of Section 5.2 to MTSR is almost immediate. For MTSR, the relaxation of the "min" condition in the non linear programming problem (5.9) yields (5.10), (5.11) and (5.20). This last equation is necessary to fulfill the flow proportions between transitions in CEQ. Once equation (5.20) is added, the B & B algorithm of subsection 5.2.2 can be applied directly to MTSR systems.



(a) (b)

Figure 5.8: Two transitions in CEQ relation and an equivalent one.

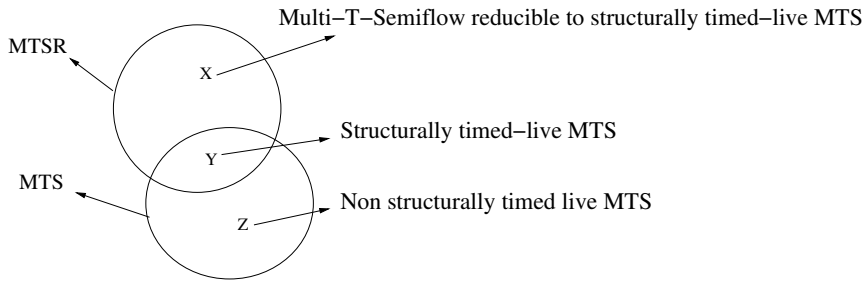


Figure 5.9: Relationship between MTS and MTSR nets. MTSR only does not include the non str. timed-live MTS nets.

In the same way, equation (5.20) must also be added to the programming problem (5.14) for MTSR systems. And with identical reasoning of subsection 5.2.5, same equations (5.15), (5.16), (5.17) and (5.19) are obtained. Every reasoning and result of subsection 5.2.5 is also directly applicable on MTSR systems.

It is interesting to remark that the class of MTSR nets offer a significant modelling power from a practical point of view. Focusing on live and bounded systems, the class of MTSR nets includes the class of equal conflict (EQ) nets [TS96], which is a superset of the classes of free-choice (FC), choice-free (CF) [TCS97], weighted T-systems (WTS) and marked graphs (MG) nets [CHEP71](being MG a generalization of PERT charts). Figure 5.10 shows the inclusion relationships among the mentioned classes.

With respect to the reachability of the upper bound computed by the LPP in (5.19) for MTSR systems (Corollary 5.6) it has to be noticed that it suffices to prove that

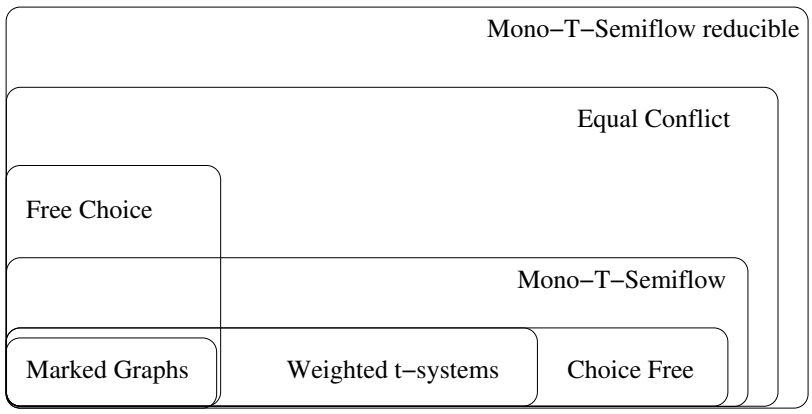


Figure 5.10: Live and bounded net classes included in the class of MTSR nets.

the minimal T-covertures contain the support of a P-semiflow. This condition is in general difficult to solve since the number of minimal T-covertures may be very large. Nevertheless, Corollary 5.6 holds for instance for *str.* lim*-live and str. bounded EQ nets* (or equivalently [STC98] EQ nets that are consistent, conservative and the rank of the token flow matrix is upper bounded by the number of conflicts). More general classes exist for which this result holds too. For instance, it holds for the net system in Figure 5.3.

5.4 Conclusions

This chapter presents a study of the throughput bounds (upper, lower and reachable bounds) in the steady state of continuous Petri net systems under infinite servers semantics. The continuized system does not always faithfully represent the original discrete system: It may happen that the performance of the discrete model is better than the performance of the continuous one. Even for the MTS subclass of net models some unexpected results may happen.

In MTS systems the vector of visit ratios does only depend on the structure of the net. Therefore, once the steady state flow of one transition is known, it is immediate to compute the flow for the rest of transitions. Upper and lower bounds for the throughput of the system in the steady state can be computed by B & B algorithms. Relaxing some conditions an upper bound can be computed by a single LPP (5.19). This LPP is based on a search for the *slowest* P-semiflow of the system and it is the continuous *version* of the one in [CS92]. It has been shown that the bound computed by the LPP will be reached iff the set of places that are determining the flow of the system in the steady state (T-coverture) contains a P-semiflow.

The class of *mono T-semiflow reducible* (MTSR) nets considers those continuous nets whose visit ratio does only depend on the structure and the internal speeds of the transitions (not on the initial marking), i.e., $\mathbf{v}^{(1)} = \mathbf{v}^{(1)}(\mathcal{N}, \boldsymbol{\lambda})$. For this class of nets the obtained results concerning the computation and reachability of the bounds for timed MTS systems are directly applicable.

Chapter 6

Observability

Summary

Observability and design of observers are topics of major importance in systems theory. This chapter shows that the conditions that characterize observability in continuous Petri nets are much less restrictive than those for general piecewise linear systems. The concept of *structural observability*, regarding to the possibility of estimating the marking of places independently of the speed of the transitions, is introduced and studied for the subclass of Join Free nets. For general nets, the observability problem is faced by computing an estimate per structural PT-set of the net and filtering those estimates that are not suitable. The design of observers is also addressed for general nets. The observer that is proposed is a piecewise linear system itself that assures the continuity of the estimate when a switch (change in the PT-set of the net) occurs. By means of the system simulation, the resulting observer allows one to estimate even the unobservable space of the net system during a given time period.

Introduction

Analysis and synthesis are two major issues of study regarding continuous Petri nets. Focusing on synthesis, a crucial topic of research is the design of control laws that drive the evolution of the system in a desired way. In order to control a dynamic system, frequently it is necessary to know its current state. To gather this information, sensors can be placed on several locations of the plant being modelled. However, it may happen that some state variables of the system cannot be directly measured by sensors. It can also happen that the cost of the sensors required to measure every state variable is prohibitive.

In a general dynamic system, under some conditions, some of the variables that cannot be directly measured can be estimated. This estimate constitutes the observation. The observability problem, i.e., the characterization of which state variables are observable and its observation, has been studied in detail in the framework of *linear* systems (see for example [Oga95, BF87]). Observability problems have also been studied in the discrete event systems setting (see for example [RTRRLM03, GS02, CGSJ03, GSJar]). For continuous linear systems, the observable subspace can be characterized algebraically. A system state estimation based on such algebraic equation can be *theoretically* obtained from the computation of the derivatives of the output signal. The estimate loses its reliability when "high" frequency noise appears in the output signal. In order to overcome this problem, linear observers came up [Lue71]. A linear observer is a linear system whose state converges asymptotically to the state of the system being observed.

This chapter is devoted to the study of observability and the design of observers for timed continuous Petri nets under infinite servers semantics [JJRS04c, JJRS04b]. It will be noticed that a great parallelism exists between the results obtained from the analysis of observability and from the observers synthesis. With regard to observability, the attention is first focused on the study of net systems without synchronizations, named Join Free (JF) systems. For this class of net systems a single differential equation system controls the behavior of the system, thus classical results on observability of linear systems apply here. The results on structural observability for JF systems that are presented in this chapter can be extended to other linear systems. Afterwards, general systems including synchronizations will be considered. Some particular features of continuous Petri nets allow one to extract less restrictive observability conditions than the ones known for regular piecewise linear systems.

A Luenberger's observer will be taken into account for each possible differential equation system (PT-set). Again, the intrinsic features of continuous Petri nets will allow one to filter non-feasible observer's estimates, and to describe the conditions that permit one to design a switching observer that will converge to the state system.

The structure of the chapter is the following: In Section 6.1 the observability

problem for continuous Petri nets is stated in a similar way to the observability problem for linear systems. Section 6.2 concentrates on JF systems. For this type of systems, structural conditions of observability are obtained from the output of a fix point algorithm. Section 6.3 is devoted to the study of the problem for general (not only JF) Petri net systems. Section 6.4 shows how a set of linear observers can be created for a net system and the different classes of non-suitable observers' estimates that can appear. Section 6.5 proposes an observer that uses a filter for the estimates and the simulation of the system.

6.1 Observability: Problem Statement

Let us consider first linear time invariant systems, for which observability has been thoroughly studied [Lue71, Oga95, BF87]. An unforced linear system (i.e., without inputs) is usually expressed by equations $\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}, \mathbf{y} = \mathbf{S} \cdot \mathbf{x}$ where $\mathbf{x}$ is the state of the system and $\mathbf{y}$ is the output, i.e., the set of measured variables. The state space is denoted as $\mathbf{X}$. Knowing the matrices $\mathbf{A}$ and $\mathbf{S}$ and being able to watch the evolution of $\mathbf{y}$, a linear system is said to be observable iff it is possible to compute its initial state, $\mathbf{x}(t_0)$ (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 well-known observability criterion exists that allows one to decide whether a continuous (deterministic) time linear system is observable or not [Oga95, BF87]. Besides, several approaches exist to compute the initial state of continuous time linear system that is observable. Nevertheless, in order to simplify the presentation of the results and make them more intuitive, the evolution of the systems will be expressed in discrete time (continuous time will be used only when dealing with structural observability, in Section 6.2) .

Given a linear system of dimension n expressed in discrete time, $\mathbf{x}(k+1) = \mathbf{F} \cdot \mathbf{x}(k)$, $\mathbf{y}(k) = \mathbf{S} \cdot \mathbf{x}(k)$ the output of the system in the first $n-1$ periods is:

$$\begin{pmatrix} \mathbf{y}(0) \\ \mathbf{y}(1) \\ \mathbf{y}(2) \\ .. \\ \mathbf{y}(n-1) \end{pmatrix} = \begin{pmatrix} \mathbf{S} \\ \mathbf{S} \cdot \mathbf{F} \\ \mathbf{S} \cdot \mathbf{F}^2 \\ .. \\ \mathbf{S} \cdot \mathbf{F}^{n-1} \end{pmatrix} \cdot \mathbf{x}_0 = \vartheta \cdot \mathbf{x}_0 \tag{6.1}$$

The matrix ϑ is called *observability matrix* [Lue71, Oga95, BF87]. The linear system is observable iff ϑ has full rank. For a non observable system it is possible to decompose the state space $\mathbf{X}$ into two subspaces: the *observable subspace*, $\mathbf{X}_o$, and the *non observable subspace*, $\mathbf{X}_{no}$. It can be verified that $\mathbf{X}_{no}$ is the kernel of ϑ, i.e., $\vartheta \cdot \mathbf{X}_{no} = \mathbf{0}$, because it does not have any influence on the vector of outputs.

Let us now consider timed continuous Petri net systems. As it has been seen, the evolution of a Petri net system is ruled by a set of switching linear systems, each one associated to a PT-set (see Subsection 1.3.2), where the state vector is the marking of the net, $\mathbf{m}$. Every linear system $\Sigma_i : \dot{\mathbf{m}} = \mathbf{A}_i \cdot \mathbf{m}$ associated to a PT-set of the Petri net can be discretized in time. The associated discrete time system can be written as $\Sigma_i^d : \mathbf{m}(k+1) = \mathbf{F}_i \cdot \mathbf{m}(k)$, with $\mathbf{F}_i = e^{\mathbf{A}_i \cdot \delta}$ where δ is the time period. The output of the net system is given by $\mathbf{y} = \mathbf{S} \cdot \mathbf{m}$. Here it will be assumed that each place is either *measured* or *unmeasured*. It will be said that a place p_i is measured iff there exists a row j in $\mathbf{S}$ such that $\mathbf{S}(j,i) \neq 0$ and $\mathbf{S}(j,k) = 0$ for every $k \neq i$.

Let us define the concept of observability for a continuous Petri net system:

Definition 6.1. Let $\mathcal{N}$ be a continuous Petri net system, $\boldsymbol{\lambda}$ the internal speeds of the transitions, and $\mathcal{D}$ the set of measured places.

- A place $p \in P$ is *observable* from $\mathcal{D}$ iff it is possible to compute its initial marking $\mathbf{m_0}[p] = \mathbf{m}(\tau_0)[p]$ by measuring the marking evolution of the places in $\mathcal{D}$.
- $\mathcal{N}$ is *observable* from $\mathcal{D}$ iff every place $p \in P$ is *observable*.

Applying [Oga95, BF87] an observability criterion is immediately deduced:

Property 6.2. Given a Petri net system and $\Sigma_i^d : \mathbf{m}(k+1) = \mathbf{F}_i \cdot \mathbf{m}(k)$ the linear system associated to PT-set i. The PT-set i is observable iff its associated observability matrix ϑ_i has full rank.

Clearly, when the net system is ruled by an observable PT-set the marking of all the places can be computed through Equation 6.1.

For a general PT-set, the places not in the PT-set can be considered as *timed-implicit* and do not play any role in the dynamics of the system. Consequently, no information about the marking of the timed-implicit places can be inferred from the marking of the places in the PT-set. Therefore, if a PT-set is wanted to be observable, the only way to compute the marking of the timed-implicit places is to take them directly in the output matrix $\mathbf{S}$.

Notice that every $\mathbf{F}_i$ is an exponential matrix and therefore it can be inverted. Hence continuous Petri net systems can be simulated backwards if the actual marking is known. Observe that the PT-set changes can be detected also from the backwards simulation. This implies that if the marking of the system at a given instant is known then the marking of the system at any previous instant can be computed.

6.2 Observability in Join Free Systems

As already pointed out, in continuous JF systems the evolution of the marking can be modelled as $\dot{\mathbf{m}} = \mathbf{A} \cdot \mathbf{m}$ (or $\mathbf{m}(k+1) = \mathbf{F} \cdot \mathbf{m}(k)$ if time is discrete or discretized). As

in Property 6.2, the existing observability criterion for linear systems can be directly applied. In this section, an effort is made to extract an observability criterion that is independent of the internal speeds of the transitions, i.e., vector $\boldsymbol{\lambda}$.

6.2.1 Structural Observability

Definition 6.3. Let $\mathcal{N}$ be a continuous Petri net and $\mathcal{D}$ the set of measured places of the system:

- Place p is *structurally observable* from $\mathcal{D}$ iff it is observable from $\mathcal{D}$ for any $\boldsymbol{\lambda} > 0$.
- $\mathcal{N}$ is *structurally observable* from $\mathcal{D}$ iff every place p is *structurally observable*.

In other words, structural observability looks for observability for any $\boldsymbol{\lambda}$, like structural boundedness looks for boundedness for any $\mathbf{m_0}$. As an example, let us suppose that the only measured place of the system in Figure 6.1 is p_3 and that the vector $\boldsymbol{\lambda}$ is known. The variation, i.e., the derivative, of the marking of a place is given by the difference between its input and output flows. For p_3, one has $\dot{\mathbf{m}}[p_3] = \mathbf{f}_2 - \mathbf{f}_3$ where: $\mathbf{f}_2 = \boldsymbol{\lambda}[t_2] \cdot \mathbf{m}[p_2]$ and $\mathbf{f}_3 = \boldsymbol{\lambda}[t_3] \cdot \mathbf{m}[p_3]$, and so $\mathbf{m}[p_2] = (\dot{\mathbf{m}}[p_3] + \boldsymbol{\lambda}[t_3] \cdot \mathbf{m}[p_3])/\boldsymbol{\lambda}[t_2]$. Therefore, from the evolution of $\mathbf{m}[p_3]$, $\mathbf{m}[p_2]$ can be computed. Furthermore, it holds $\dot{\mathbf{m}}[p_2] = \mathbf{f}_1 - \mathbf{f}_2$ and $\mathbf{f}_1 = \boldsymbol{\lambda}[t_2] \cdot \mathbf{m}[p_1]$. Thus, being $\mathbf{m}[p_2]$ computable, $\mathbf{m}[p_1]$ can also be computed. This procedure can be carried out whatever the value of $\boldsymbol{\lambda}$ is, i.e. this net is structurally observable.

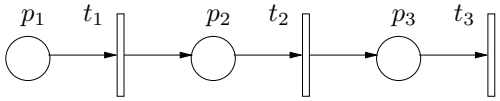


Figure 6.1: A JF net system whose marking can be computed from the observation of p_3.

This result can be easily generalized:

Proposition 6.4. Let $\mathcal{N}$ be a continuous JF Petri net and $\mathcal{D}$ the set of measured places. Let p be a place such that a path from p to $\mathcal{D}$ exists in which all the places have only one input transition (i.e., it is attribution free). Then, p is structurally observable.

Proof. In a JF system, the output flow of a place p is proportional to its marking, $\mathbf{f}_{out}[p] = \sum_{t \in p^\bullet} \mathbf{Pre}[p,t] \cdot \boldsymbol{\lambda}[t] \cdot \text{enab}(t, \mathbf{m}) = \sum_{t \in p^\bullet} \boldsymbol{\lambda}[t] \cdot \mathbf{m}[p]$. From $\dot{\mathbf{m}}[p] = \mathbf{f}_{in}[p] - \mathbf{f}_{out}[p]$, if $\mathbf{m}[p]$, and $\dot{\mathbf{m}}[p]$ are known, the total input flow of p can be obtained. If place p has only one input transition it is easy to obtain the marking of the input place of that transition, $\mathbf{f}_{in}[p]/\mathbf{Post}[p, {}^\bullet p] = \boldsymbol{\lambda}[{}^\bullet p] \cdot \mathbf{m}[{}^\bullet({}^\bullet p)]/\mathbf{Pre}[{}^\bullet({}^\bullet p), {}^\bullet p]$. □

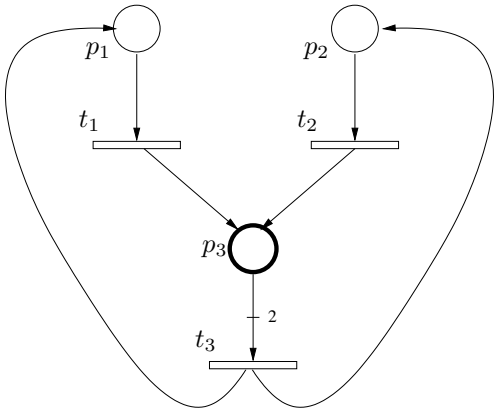


Figure 6.2: A JF net system whose marking cannot be computed from the observation of p_3 if $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$, because t_1 and t_2 make an attribution to p_3.

However, for the system in Figure 6.2, if p_3 is measured, it is not possible to observe p_1 or p_2 if $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2]$ (it can be seen that the observability matrix does not have full rank if this condition holds). Intuitively, if the flow of both transitions comes with "the same speed" it cannot be decided how much comes from each source. Hence, this net is not structurally observable. However, it is observable when $\boldsymbol{\lambda}[t_1] \neq \boldsymbol{\lambda}[t_2]$.

On the contrary, from the knowledge of the marking of place p it is not possible to compute the marking of the place(s) $(p^\bullet)^\bullet$: Let $p' \in (p^\bullet)^\bullet$, and assume to simplify that $p^\bullet = {}^\bullet p' = t$. To compute the marking of p' it would be necessary to solve $\dot{\mathbf{m}}[p'] = \mathbf{f}_{out}[p] \cdot \mathbf{Post}[p', t]/\mathbf{Pre}[p, t] - \mathbf{f}_{out}[p']$. If the initial marking of p' is not known that equation cannot be solved whatever the value of $\mathbf{Pre}, \mathbf{Post}$ and $\boldsymbol{\lambda}$ are. This means that from the measured places at most the marking of the supplying places can be inferred.

Moreover, it can be proved that the set of places whose marking can be computed does not depend on the "output" of the measured places.

Proposition 6.5. Let $\mathcal{N}$ be a continuous JF Petri net and $\mathcal{D}$ the set of measured places. The observable subspace (and so the set of structurally observable places) neither depends on the output arc weights of the measured places, $\mathbf{Pre}[p, T]\ \forall p \in \mathcal{D}$, nor on the firing speeds of their output transitions, $\boldsymbol{\lambda}[t]\ \forall\ t \in \mathcal{D}^\bullet$, nor on the output arc weights of their output transitions, $\mathbf{Post}[p, T]\ \forall\ p \in (\mathcal{D}^\bullet)^\bullet$.

Applying Proposition 6.5 to the system in Figure 6.3(a) with p_4 as the only measured place, the observable subspace of the system does not depend on the values of $r, s, t, q, \boldsymbol{\lambda}[t_4]$. Even after removing the input/output arcs of transition t_4 as in Figure 6.3(b) (this is equivalent to $\boldsymbol{\lambda}[t_4] = 0$), the obtained system is identical to the

original one in terms of observability.

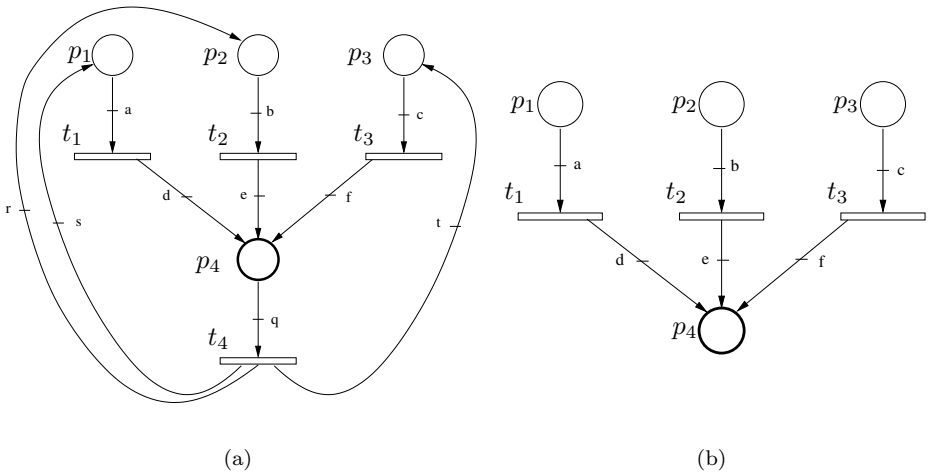


Figure 6.3: Two net systems with identical observable subspaces if the only measured place is p_4.

According to Proposition 6.5 the output transitions of the measured places can be removed (by setting their $\boldsymbol{\lambda}$ to zero) without affecting the observable subspace of the system. Therefore:

Corollary 6.6. Let $\mathcal{N}$ be a continuous Petri net and $\mathcal{D}$ the set of measured places. If a place p is structurally observable then a forward path from p to $\mathcal{D}$ exists.

As an example, let us consider that the only measured place in Figure 6.1 is place p_2. Using the derivative of the marking of p_2 the marking of p_1 can be computed. However, there exists no forward path going from p_3 to p_2 and therefore, according to Corollary 6.6, p_3 is not structurally observable. Intuitively, the marking of p_3 cannot be deduced from its input flow.

6.2.2 Computation Algorithm

A similar approach to the one taken to observe p_1 and p_2 in the system in Figure 6.1 can be used to observe p_1 and p_2 in the system in Figure 6.4, where the measured places are p_3, p_4 and p_5. Let us consider a matrix $\mathbf{Post}^u \in \mathbb{R}^{|P|\times|T|}$ containing only the output arc weights of the transitions whose flow is, "in principle", unknown, i.e., the marking of their input places is not known. More formally, for the iterative algorithm that will be proposed, $\mathbf{Post}^u[i,j] = 0$ if the marking of the place ${}^\bullet t_j$ is

measured or has been computed in previous iterations, and $\mathbf{Post}^u[i,j] = \mathbf{Post}[i,j]$ otherwise.

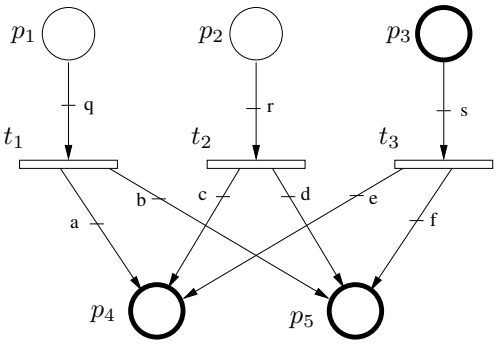


Figure 6.4: A JF system whose marking is computable from the evolution of p_3, p_4 and p_5.

Here, the first three rows (that correspond to places p_1, p_2 and p_3) of $\mathbf{Post}^u$ are zeros and the forth and fifth rows (that correspond to places p_4 and p_5) are $(a\ c\ 0)$ and $(b\ d\ 0)$, respectively. The marking evolution of places p_4 and p_5 is known (because they are measured) and here it is equal to their input flow. Subtracting the flow coming from p_3, $\mathbf{f}_{i4}^{p_3}$ and $\mathbf{f}_{i5}^{p_3}$, the flow coming from the unknown places p_1 and p_2 will be obtained:

$$\begin{pmatrix} \dot{\mathbf{m}}[p_4] - \mathbf{f}_{i4}^{p_3} \\ \dot{\mathbf{m}}[p_5] - \mathbf{f}_{i5}^{p_3} \end{pmatrix} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\lambda}[t_1] \cdot \dfrac{\mathbf{m}[p_1]}{q} \\ \boldsymbol{\lambda}[t_2] \cdot \dfrac{\mathbf{m}[p_2]}{r} \end{pmatrix}$$

Hence, if the matrix $(a\ c; b\ d)$ has full rank it will be possible to compute the markings of p_1 and p_2 independently of the $\boldsymbol{\lambda}$ of the transitions.

The procedure developed for the above examples can be generalized leading to a fix point algorithm. The goal of the algorithm is, given a set of measured places, $\mathcal{D}$, to deduce which places of the net system can be observed for whatever value of $\boldsymbol{\lambda}$. The basis of such iterative algorithm is to look for sets of places whose marking has been computed in previous iterations and such that the matrix composed of their input arcs weights has full rank. If such a set exists, then it is possible to compute the marking of the supplying places.

Given a set of places $\mathcal{H}$, $\mathbf{Post}_{\mathcal{H}}^{u}$ denotes a matrix composed by the rows of $\mathbf{Post}^u$ corresponding to the places in $\mathcal{H}$, and whose null columns have been removed. The input data of the algorithm are the net structure, $\mathcal{N}$, and the set of measured places, $\mathcal{D}$. The output of the algorithm is the set of places, $\mathcal{Q}$, that are observable for every $\boldsymbol{\lambda}$. At a given iteration, $\mathcal{Q}$ stores the set of places whose markings are known to be computed till that instant.

Algorithm 6.7 (Algorithm for structural observability).
Input: $(\mathcal{N}, \mathcal{D})$
Output: $\mathcal{Q}$ % places that can be observed $\forall\ \boldsymbol{\lambda} > 0$
Begin
$\mathcal{Q} := \mathcal{D}$
Compute $\mathbf{Post}^u$
While $\exists\ \mathcal{H} \subseteq \mathcal{Q}$, such that $^\bullet(^\bullet\mathcal{H}) \not\subset \mathcal{Q}$ and $\mathbf{Post}^u_{\mathcal{H}}$ has full rank **do**
$\mathcal{Q} := \mathcal{Q} \cup {}^\bullet(^\bullet H)$
Compute $\mathbf{Post}^u$ according to $\mathcal{Q}$
End_While
End

The following statements establish sufficient conditions for structural observability:

Proposition 6.8. Let $\mathcal{N}$ be a JF net, $\mathcal{D}$ the set of measured places and $\mathcal{Q}$ the output of the Algorithm 6.7 applied on $(\mathcal{N}, \mathcal{D})$:

- Every place $p \in \mathcal{Q}$ is structurally observable.
- If $\mathcal{Q} = P$ (the set of places of $\mathcal{N}$) then the net is structurally observable.

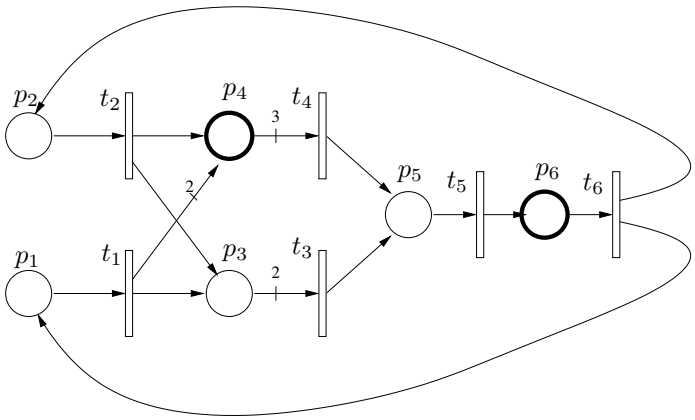


Figure 6.5: A JF net system that is structurally observable.

If the measured places of the system in Figure 6.5 are p_4 and p_6, the first iteration of Algorithm 6.7 on this system includes p_5 in the set of observable places. The second iteration includes p_3 and the third and last iteration includes p_1 and p_2. Therefore, it can be concluded that the whole net is structurally observable.

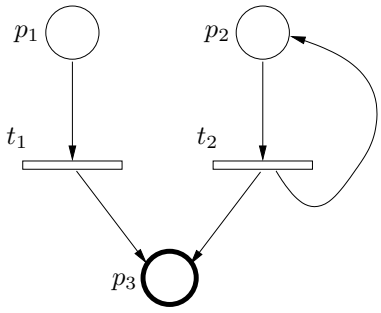


Figure 6.6: A JF system for which Algorithm 6.7 does not conclude that it is structurally observable.

Unfortunately, the conditions given in Proposition 6.8 are only sufficient for structural observability. The execution of Algorithm 6.7 on the system in Figure 6.6, whose only measured place is p_3, yields $\mathcal{Q} = p_3$, i.e., Proposition 6.8 cannot decide whether the system is structurally observable or not. Nevertheless, the observability matrix, ϑ, for that system is $\vartheta = (0\ 0\ 1; \boldsymbol{\lambda}[t_1]\ \boldsymbol{\lambda}[t_2]\ 0; -\boldsymbol{\lambda}[t_1]^2\ 0\ 0)$ which has full rank for every $\boldsymbol{\lambda} > 0$. Therefore the system is structurally observable.

Although a formal proof is missing, after testing a wide variety of examples, it seems reasonable to think that the condition on Proposition 6.8 is necessary and sufficient for those JF systems that are conservative.

6.3 Observability in General Net Systems

The goal of this Section is the study of the conditions under which the initial PT-set as well as the initial marking can be unequivocally determined.

A way to face this problem consists of computing an estimate for every structural PT-set of the net. For simplicity, let the estimates be obtained by means of Equation 6.1 defined for $n-1$ consecutive periods. Theoretically and assuming that there is no noise, time discretization (δ) can be done as small as desired. It can be assumed that, for a small enough δ, no switch between PT-sets takes place in the first $n-1$ periods. The computed estimates can be used to filter those PT-sets that for sure are not ruling the evolution of the system. If only one PT-set remains, then the system evolves according to it.

The sorts of non suitable estimates that allow one to filter a PT-set are:

- *Infeasible* estimates: No solution of Equation 6.1
- *Incoherent* estimates: The PT-set of the estimate is not the one for which it was computed

- *Suspicious* estimates: The estimate belongs simultaneously to several PT-sets.

6.3.1 Infeasible and Suspicious Estimates

Let us show through an example how infeasible and suspicious estimates can be used to filter PT-sets.

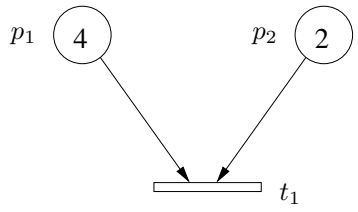


Figure 6.7: A simple synchronization with two input places.

Consider a system composed of a single synchronization with two input places p_1 and p_2 (see Figure 6.7). The net has two structural PT-sets, either $W_1 = \{(p_1, t_1)\}$ or $W_2 = \{(p_2, t_1)\}$. If the time period is one time unit, the evolution of the system according to PT-set W_1 is ruled by the matrix $\mathbf{F}_1 = (e^{-1} \;\; 0; e^{-1} - 1 \;\; 1)$. The system matrix for PT-set W_2 is $\mathbf{F}_2 = (1 \;\; e^{-1} - 1; 0 \;\; e^{-1})$. Considering that the initial marking is $\mathbf{m_0} = (4; 2)$, the initial PT-set for the system is W_2, and after one time unit the marking will be $\mathbf{m}(\tau = 1) = (2 \cdot e^{-1} + 2; 2 \cdot e^{-1})$.

As external agents of the system 4 cases will be considered (see Figure 6.8): The output of the system, i.e., measured place, can be either p_1 or p_2; the PT-set that is assumed to be ruling the net system can be either W_1 or W_2.

In the first two cases $\mathbf{m}[p_1]$ is the output of the system ($\mathbf{y} = (4; 2 \cdot e^{-1} + 2)$):

Case 1: W_2 is assumed to be the PT-set. For this case $\vartheta = (1 \; 0; 1 \; e^{-1} - 1)$ whose rank is 2. Using Equation 6.1 the initial marking $\mathbf{m_0} = (4; 2)$ is obtained.

Case 2: W_1 is assumed to be the PT-set. The observability matrix is $\vartheta = (1 \; 0; e^{-1} \; 0)$. Equation 6.1 has no solution.

In this way, by means of an "infeasible" estimate (case 2), it has been detected that the PT-set of the system is W_2.

If p_2 is measured, ($\mathbf{y} = (2; 2 \cdot e^{-1})$), the two cases are:

Case 3: W_2 is assumed to be the PT-set. $\vartheta = (0 \; 1; 0 \; e^{-1})$, which is not a full rank matrix. The observable subspace is in this case $\mathbf{m}[p_2]$, i.e., only the marking of p_2 is known. The application of Equation 6.1 yields $\mathbf{m_0}[p_2] = 2$.

Case 4: W_1 is assumed to be the PT-set. The observability matrix is $\vartheta = (0 \; 1; e^{-1} - 1 \; 1)$ which has full rank. The solution for Equation 6.1 is $\mathbf{m_0} = (2 \; 2)$, different from the real initial marking of the system.

System output / Estimate PT−set	p1 S=(1 0)	p2 S=(0 1)
W2	case 1 Right estimation of m1 and m2	case 3 Right estimation of m2
W1	case 2 Infeasible estimation	case 4 Suspicious estimation (m1 = m2)

Figure 6.8: The four possible cases for an estimate for the system in Figure 6.7.

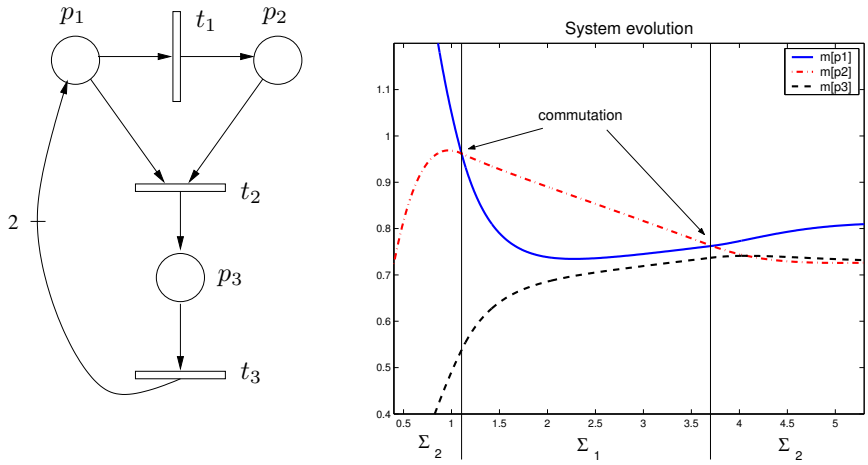
The at first glance surprising result yielded in Case 4 will be obtained for any initial marking of p_1 greater than or equal to 2. The reason for this phenomenon is that the output of the system, $\mathbf{m}[p_2]$, is evolving according to the flow of the transition t_1 that depends only on $\mathbf{m}[p_2]$. The estimator "thinks" that the flow of the transition is ruled by p_1 (W_1 is assumed), so the only way in which p_2 can evolve according to the output $\mathbf{y}$ is assigning the same initial marking to both places.

An initial marking $\mathbf{m_0} = (2\ 2)$ would mean that at the beginning the system is in both PT-sets, W_1 and W_2. If it is assumed that the initial marking was really (2 2) both PT-sets (case 3 and case 4) would correctly estimate here the marking of p_2. The PT-set W_1 can also estimate the marking of p_1, but it is not possible to know whether this estimate is correct since the same estimate would have been obtained for any $\mathbf{m_0}[p_1] \geq 2$. Therefore, it seems safer to stick to the PT-set W_2, even if it might mean losing some information. Those estimates that belong to several PT-sets will be considered "suspicious estimates". Notice that the only non desirable effect that may happen after filtering suspicious estimates is that some information (in case 4 the estimate of p_1) is lost if the system was really in several PT-sets. However, for sure the estimate that is not filtered (case 3) is correct.

6.3.2 Incoherent Estimates

Another rule to filter a PT-set is that the estimates should be *coherent* with the PT-set for which they are computed. In other words, it does not make sense to consider an estimate that assigns a greater marking to p_1 than to p_2, if the PT-set for which it is computed happens when $\mathbf{m}[p_1] \leq \mathbf{m}[p_2]$.

By means of Equation 6.1, it is possible to compute the set of estimates for a given PT-set. If the matrix ϑ has full rank, then only one estimate is possible for the PT-set. Otherwise, a set of possible estimates appears. In order to avoid those estimates that are not coherent with the PT-set, a set of inequalities may be added to Equation 6.1.



(a) (b)

Figure 6.9: Marking evolution of a system with two PT-sets.

As an example, suppose that one is interested in computing an estimate for the system in Figure 6.9(a). That net has two structural PT-sets: it will be said that the system is in PT-set W_1 if $\mathbf{m}[p_1] \leq \mathbf{m}[p_2]$ and the system is in PT-set W_2 if $\mathbf{m}[p_1] \geq \mathbf{m}[p_2]$. In the case that $\mathbf{m}[p_1] = \mathbf{m}[p_2]$, the system is considered to be in both PT-sets simultaneously. Two estimates will be computed for this system, one per PT-set. The estimate corresponding to PT-set W_1, $\hat{\mathbf{m}}_0^{(1)}$, (W_2, $\hat{\mathbf{m}}_0^{(2)}$) has to be solution of Equation 6.1 with ϑ computed for the linear system associated to the PT-set and $\hat{\mathbf{m}}_0^{(1)}$ ($\hat{\mathbf{m}}_0^{(2)}$) has to fulfill $\hat{\mathbf{m}}_0^{(1)}[p_1] < \hat{\mathbf{m}}_0^{(1)}[p_2]$ ($\hat{\mathbf{m}}_0^{(2)}[p_1] > \hat{\mathbf{m}}_0^{(2)}[p_2]$). The use of strict inequalities allows one to filter also suspicious estimates like the one shown in Subsection 6.3.1. If there were no solution with strict inequalities, equalities would have to be added taking care of the suspicious cases.

6.3.3 Deciding on Observability

Let us observe the output of the system in Figure 6.9(a) during three time periods in order to have enough output information to use Equation 6.1. Let us assume

that no change of PT-set has taken place during these three time periods. After the observation, two equation systems, E_1 and E_2, can be defined to compute an estimate for the initial marking. The system E_1 (resp. E_2) contains Equation 6.1 with ϑ_1 (resp. ϑ_2) and the set of inequalities that defines the PT-set W_1, i.e., $\hat{\mathbf{m}}_0^{(1)}[p_1] < \hat{\mathbf{m}}_0^{(1)}[p_2]$ (resp. W_2, $\hat{\mathbf{m}}_0^{(2)}[p_1] > \hat{\mathbf{m}}_0^{(2)}[p_2]$). If only one of those equation systems has solution, that equation system corresponds to the initial PT-set of the net system. If both equation systems, E_1 and E_2 have solution, it is not possible to decide the initial PT-set of the system. The case in which none of the equation systems has solution happens when the initial marking of the system is in both PT-sets at the same time, i.e., $\mathbf{m}_0[p_1] = \mathbf{m}_0[p_2]$. In this case, the inequalities in E_1 and E_2 must be substituted by $\hat{\mathbf{m}}_0^{(1)}[p_1] = \hat{\mathbf{m}}_0^{(1)}[p_2]$. Any solution of E_1 or E_2 is a suitable estimate for the initial marking.

For a general Petri net system with k structural PT-sets, a set of equation systems, $E_1 \dots E_k$, can be defined. Each E_i contains Equation 6.1 with ϑ_i and the set of inequalities that defines the PT-set.

Proposition 6.9. Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ be a continuous system, whose initial marking, $\mathbf{m_0}$, is unknown but belongs to only one PT-set. Let $\mathbf{S}$ be the output matrix of the system and E_i the set of equations associated to the i-th PT-set.

Then the PT-set of $\mathbf{m_0}$ can be determined before a switch to another PT-set happens iff only one system E_i, $1 \leq i \leq k$ has solution.

Proof.
($\Rightarrow$) Let us assume that more than one system, E_i and E_j, have solution. This would imply that at least two estimates for the initial marking exist, one for E_i and one for E_j. Taking those estimates as initial marking, the constraints for the PT-sets that E_i and E_j represent are verified, and the system output evolves according to the dynamics of E_i and E_j. This means, that both estimates are feasible, and so the initial PT-set of the system cannot be determined.
($\Leftarrow$) If only one system E_i has solution, it means that from the initial marking the output of the system can only evolve according to the equations in E_i. □

In relation to general piecewise linear systems, determining the PT-set of a continuous Petri net is equivalent to determining the linear system that at a certain moment rules the evolution of the piecewise linear system. However, the condition that establishes whether it is possible to determine the linear system is much harder in general piecewise linear systems: in contrast to the condition in Proposition 6.9, it is required that the *joint observability matrix* of every couple of linear systems has full rank [VCS02, VCSS03]. This difference is due to the fact that in Petri nets the PT-set depends only on the marking (and the net structure).

For a system $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$, the verification of the condition in Proposition 6.9 implies that the initial PT-set can be determined. However, it does not imply that the initial

marking can be obtained. This happens when there exists only one system E_i that has solution, but the solution is not unique, i.e., there exists a non observable subspace. Assume that $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ switches to a PT-set that corresponds to an observable linear system. By using Equation 6.1 during the evolution of the system in this new PT-set, the complete marking can be computed. Once the complete marking of the system is known, the system can be "simulated" backwards, see Section 6.1. Since timing is deterministic, a backwards simulation till the initial time yields the initial marking, i.e., the system is observable.

Proposition 6.10. Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ be a continuous system and $\mathbf{S}$ the output matrix. If the evolution of the marking of $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m_0} \rangle$ passes through an observable PT-set then the system is observable.

In other words, Proposition 6.10 implies that having a period in the evolution of the system during which the PT-set allows one to compute the complete current marking is enough to determine the initial marking.

6.4 Observers and estimates

The previous section shows how an estimate can be computed by using Equation 6.1. The main drawback of that method is that it is very sensitive to the noise that may appear in the output $\mathbf{y}$. In order to overcome the problem of noise, observers are introduced. Basically, an observer is a dynamical system whose input is the output of the system to be observed. The state of an observer is the estimate for the system to be observed. It will be shown that a great parallelism exists between algebraically computed estimates and observers' estimates. A well designed observer should converge asymptotically to the real state of the observed system. For linear systems, Luenberger's observers [Lue71, Oga95] are widely used. A Luenberger observer for a Petri net with a single PT-set can be expressed as: $\dot{\tilde{\mathbf{m}}} = \mathbf{A} \cdot \tilde{\mathbf{m}} + \mathbf{K} \cdot (\mathbf{y} - \mathbf{S} \cdot \tilde{\mathbf{m}})$ where $\tilde{\mathbf{m}}$ is the marking estimate, $\mathbf{A}$ and $\mathbf{S}$ (see Section 6.1) are the matrices defining the evolution of the system marking and its output in continuous time, $\mathbf{y}$ is the output of the system and $\mathbf{K}$ is a design matrix of parameters. The eigenvalues of the observer can be chosen arbitrarily, by means of $\mathbf{K}$, iff the system to be observed is observable. If the eigenvalues of the observer are appropriately chosen then the estimate will converge asymptotically to the marking of the system. In the case that the system is not observable, an observer to estimate the observable subspace, $\mathbf{X}_o$, can be designed.

The reliability of an estimate can be measured by means of a *residual* [BBDSV02]. Let us define a norm $||\cdot||$ as $||\mathbf{x}|| = |\mathbf{x}_1| + \ldots + |\mathbf{x}_n|$. The residual at a given instant, $r(\tau)$, is the distance between the output of the system and the output that the observer's estimate, $\tilde{\mathbf{m}}(\tau)$, yields, i.e., $r = ||\mathbf{S} \cdot \tilde{\mathbf{m}}(\tau) - \mathbf{y}(\tau)||$.

6.4.1 Filtering estimates

One (Luenberger) linear observer will be designed per PT-set of the Petri net system. The designed observers will be launched simultaneously, and each one of them will yield an estimate. Some estimates may not be suitable for the PT-sets for which they are computed. Such estimates cannot represent the marking of the system and must be filtered. Three conditions will be presented that the estimates of the observers have to fulfill in order to be suitable: 1) the residual must tend to zero; 2) the estimates of places in synchronization have to be coherent with the PT-set for which they are computed; 3) the estimate must not be *suspicious*, i.e., it must not belong to several PT-sets at the same time.

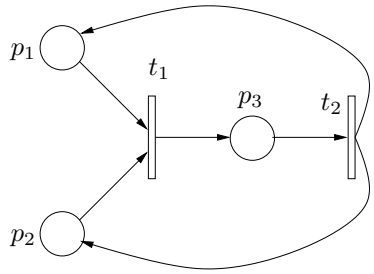


Figure 6.10: A simple general Petri net system with two PT-sets.

Let us consider the system in Figure 6.10 with $\boldsymbol{\lambda} = (1\ 1)$ to show the behavior of the observers and their estimates under different conditions. The net has two PT-sets: $W_1 = \{(p_1, t_1), (p_3, t_2)\}$ and $W_2 = \{(p_2, t_1), (p_3, t_2)\}$. The system has a single T-semiflow, (1 1). Hence, in the steady state the flow of both transitions is the same. Since the net has two PT-sets, two linear observers can be designed.

Residuals

Let us consider the observer designed for PT-set W_1. Such observer assumes that $\mathbf{m}[p_1] \leq \mathbf{m}[p_2]$ and so the system matrix in continuous time is $\mathbf{A} = (-1\ 0\ 1; -1\ 0\ 1; 1\ 0\ -1)$. Let us assume that the output of the system is the marking of places p_1 and p_3, i.e., $\mathbf{S} = (1\ 0\ 0; 0\ 0\ 1)$. Under this conditions the observable subspace, $\mathbf{X}_o$, corresponds just to the marking of places p_1 and p_3. That is, p_2 is *timed-implicit*, i.e., its marking is not giving the minimum in the expression for the enabling degree, and cannot be observed (in this case it is also implicit and could have been removed). Therefore, the observer sees the evolution of a dynamical system ruled by matrix $\mathbf{A}' = (-1\ 1; 1\ -1)$ for places p_1 and p_3.

If the real PT-set of the system is W_1 the system will evolve to a steady state marking, $\mathbf{m}$, at which $\mathbf{m}[p_1] = \mathbf{m}[p_3] < \mathbf{m}[p_2]$. An observer with appropriate eigen-

values will asymptotically converge to an estimate marking $\tilde{\mathbf{m}}[p_1] = \tilde{\mathbf{m}}[p_3]$ and the residual will go to 0 as time increases. If the real PT-set of the system is p_2, p_3, then in the steady state $\mathbf{m}[p_2] = \mathbf{m}[p_3] < \mathbf{m}[p_1]$. It can be checked, that the observer will not reach a steady state estimate in which $\tilde{\mathbf{m}}[p_2] = \tilde{\mathbf{m}}[p_3]$ and therefore the residual will not tend to 0. In this case, the information given by the residual allows one to decide that W_1 is not the PT-set of the net system. In general, every observer's estimate whose residual is not converging to 0 has to be filtered.

Coherent Estimates

It will be said that an estimate is coherent with the PT-set for which it was computed if it belongs to that PT-set. Let us consider again the observer designed for PT-set W_1 of the system in Figure 6.10 and let now the output matrix be $\mathbf{S} = (1\ 0\ 0; 0\ 1\ 0)$. In this case the observable subspace is complete, i.e., $\mathbf{X} = \mathbf{X}_o$ and the marking of every place can be estimated. For the observer, p_2 has no influence on the dynamics of the system since it is not in W_1. In the steady state it verifies $\tilde{\mathbf{m}}[p_2] = \mathbf{m}[p_2]$. The observer *thinks* (assuming that PT-set W_1 is the real PT-set) that in the steady state $\mathbf{m}[p_1]$ has to equal $\mathbf{m}[p_3]$ so that the flow of t_1 is equal to the flow of t_2. Therefore, the system estimate converges to $\tilde{\mathbf{m}}[p_1] = \tilde{\mathbf{m}}[p_3] = \mathbf{m}[p_1]$. In this way, the residual, $r = |\tilde{\mathbf{m}}[p_1] - \mathbf{m}[p_1]| + |\tilde{\mathbf{m}}[p_2] - \mathbf{m}[p_2]|$, is always equal to 0 in the steady state, independently of the real PT-set of the system.

The same phenomenon appears in the observer for PT-set W_2. The estimate converges to $\tilde{\mathbf{m}}[p_1] = \mathbf{m}[p_1]$ and $\tilde{\mathbf{m}}[p_2] = \tilde{\mathbf{m}}[p_3] = \mathbf{m}[p_2]$, independently of the real PT-set of the system. So, the residual always converges to 0.

Therefore, residuals are not helping to decide which PT-set of the system is the correct one. In principle, both observers are equally good since both residuals tend to 0. However, in order to choose the correct one it is enough to consider the marking of the places in the synchronization. Since, in this case both places are output of the system, it can be directly decided in which of the PT-sets the system is. In a general case, the estimate of an observer that is not coherent with its PT-set has to be filtered.

Suspicious Estimates

Let us consider the system in Figure 6.10 with output matrix $\mathbf{S} = (1\ 0\ 0; 0\ 0\ 1)$. The observable subspace of the observer for PT-set $W_2 = \{(p_2, t_1), (p_3, t_2)\}$ is complete. For this observer, the marking of place p_1 does not play any role in the evolution of the system. The estimate of place p_1 will always converge to the real marking of p_1, $\tilde{\mathbf{m}}[p_1] = \mathbf{m}[p_1]$. In the steady state, the observer will equal its estimates of p_2 and p_3 in order to fire transitions t_1 and t_2 in the same amount. Since $\mathbf{m}[p_3]$ is taken as output, the estimate will converge to $\tilde{\mathbf{m}}[p_2] = \tilde{\mathbf{m}}[p_3] = \mathbf{m}[p_3]$.

Let us assume that the real PT-set of the system is PT-set W_2. Then, in the steady state $\mathbf{m}[p_1] > \mathbf{m}[p_2] = \mathbf{m}[p_3]$ and according to the above reasonings the observer's estimate will correctly converge to the real marking of the system. If the real PT-set is PT-set W_1, the marking reached in the steady state fulfills $\mathbf{m}[p_2] > \mathbf{m}[p_1] = \mathbf{m}[p_3]$, and therefore the observer will converge to an estimate marking, $\tilde{\mathbf{m}}$, such that $\tilde{\mathbf{m}}[p_1] = \tilde{\mathbf{m}}[p_2] = \tilde{\mathbf{m}}[p_3]$. This estimate is considered *suspicious* because it assigns exactly the same markings to two places in synchronization for any initial marking $\mathbf{m_0}$ of the system such that $\mathbf{m_0}[p_2] \geq \mathbf{m_0}[p_1]$.

6.4.2 Observers' steady state

Some conditions to detect non-suitable observers' estimates have just been presented. A very tight relationship can be established between these conditions and those described in Section 6.3 for the estimates computed using Equation 6.1: for example, Equation 6.1 has no solution for a given PT-set iff the observer's estimate for that PT-set has a non null residual in the steady state. In the same way, suspicious or non-coherent estimates appear according to Equation 6.1 when there exists an observer whose estimate in the steady state is suspicious or non-coherent. Unlike the estimates computed in Section 6.3, the estimate yielded by an observer becomes more reliable as time increases.

When the system enters the steady state, its marking can be considered to remain constant and observers stabilize. At this point, those estimates that generate a non null residual or are not coherent must be filtered. Suspicious estimates must also be filtered if there exists at least another estimate that is not suspicious. If there is only one observer's estimate that has not been filtered, it is associated to the real PT-set of the system in the steady state. Assuming that in the steady state the system is not in more than one PT-set, the necessary and sufficient condition that has to be verified in order to filter every but one estimate, is equivalent to that of Proposition 6.9.

Proposition 6.11. Let us assume that the steady state marking of a system belongs only to one PT-set. In the steady state only one observer's estimate is not filtered, i.e., it generates a null residual, it is coherent with its PT-set and it is not suspicious iff only one equation system $E_i,\ 1 \leq i \leq n$ as defined in Section 6.3.3 associated to PT-set i has solution.

During the transient state, the estimates given by the observers may not be very reliable since there has not been enough time to get stabilized. At a given instant in the transient, the number of observer's estimates that are coherent with their PT-set may differ from the number of E_i systems that have solution. When this happens, a way to choose an estimate consists of choosing the one with minimum residual.

6.5 Design of a switching observer

This Section shows the design of an observer based on the filter of non-suitable estimates and the simulation of the system.

6.5.1 Filter based observer

Let us consider the continuous Petri net system in Figure 6.9(a). Let the output of the system be the marking of place p_1, that is, $\mathbf{S} = (1\ 0\ 0)$. The net has two PT-sets: let one of the PT-sets be $Z_1 = \{(p_1, t_1), (p_1, t_2), (p_3, t_3)\}$ and the other be $Z_2 = \{(p_1, t_1), (p_2, t_2), (p_3, t_3)\}$. The observable subspace of the PT-set Z_1 is the marking of places p_1 and p_3, while the observable subspace of PT-set Z_2 is the marking of all the places. Let the internal speeds of the transitions be $\boldsymbol{\lambda} = (0.9\ 1\ 1)$ and the initial marking be $\mathbf{m_0} = (3\ 0\ 0)$. The marking evolution of this system is depicted in Figure 6.9(b).

One observer per PT-set will be designed: observer 1 for PT-set Z_1 and observer 2 for PT-set Z_2. Let the initial state of observer 1 be $\mathbf{e}_{01} = (1\ 2)$ and its eigenvalues be $(-12 + 2 \cdot \sqrt{3} \cdot i,\ -12 - 2 \cdot \sqrt{3} \cdot i)$. Since observer 1 can only estimate p_1 and p_3, the first component of its state vector corresponds to the estimate for $\mathbf{m}[p_1]$, and its second component to the estimate for $\mathbf{m}[p_3]$. For observer 2, let the initial state be $\mathbf{e}_{02} = (1\ 0\ 2)$ and its eigenvalues be $(-15,\ -12 + 2 \cdot \sqrt{3} \cdot i,\ -12 - 2 \cdot \sqrt{3} \cdot i)$. The evolutions of the estimates of the observers are depicted in Figures 6.11 and 6.12.

The estimate of observer 1 gets quite close to the real marking of the system when it is in PT-set Z_1. At time $\tau = 3.7$ the system switches to PT-set Z_2 and the estimate for the marking of place p_3 moves away from the real marking. Similarly, the estimate of observer 2 gets very close to the marking of the system before it switches to PT-set Z_1 at time $\tau = 1.1$. As soon as the system switches, the observer loses the goodness of the estimate. When the system switches back to PT-set Z_2, the estimate approaches back quickly to the marking of the system.

After launching the observers for the PT-sets, a criterion must be adopted to decide which the best observer's estimate is. First, let us just filter the observer's estimate that has the greatest residual, see Figure 6.13. Before the first switch, observer 2 is chosen. After the switch, some time elapses till the residual of observer 2 becomes greater than the residual of observer 1. When this happens, the estimate of observer 2 is filtered. A similar phenomenon can be seen when the system switches from PT-set Z_1 to PT-set Z_2: after a little time the estimate of observer 2 becomes smaller than the estimate of observer 1.

Notice that from the first system switch till the switch of observers, observer 2 has the minimum residual. However, it is not coherent with the PT-set for which it was designed, since $\tilde{\mathbf{m}}[p_1] < \tilde{\mathbf{m}}[p_2]$. Let us improve the estimate given by the observers

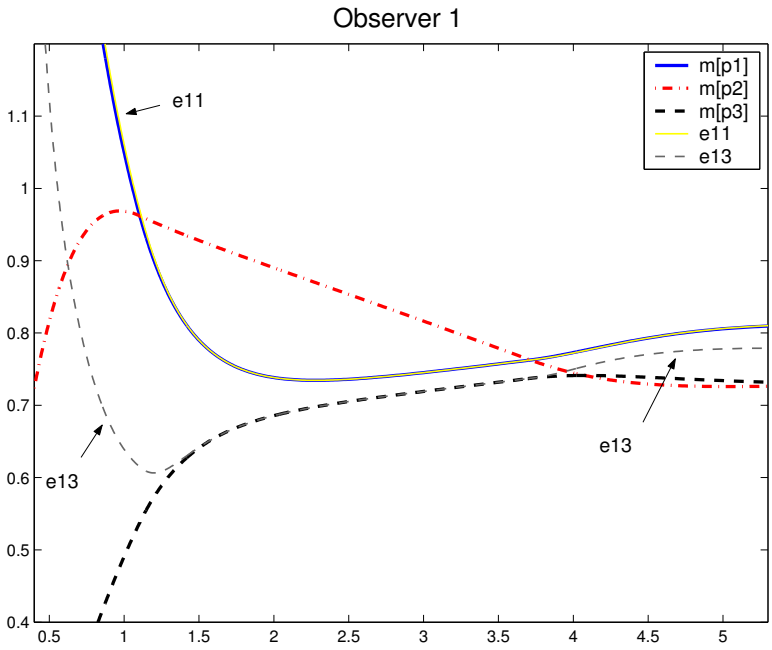


Figure 6.11: Evolution of observer 1, ($e11$, $e13$) is the estimate for ($\mathbf{m}[p_1]$, $\mathbf{m}[p_3]$).

by filtering those estimates that are not coherent with their PT-sets, see Figure 6.14. In this way, the first switch is immediately detected.

6.5.2 Improving the observer's estimate

The filter described in the previous Subsection allows one to eliminate non-suitable estimates, i.e., infeasible, non-coherent and suspicious estimates. However, the resulting estimate can still be improved by taking into account some considerations. Let us have a look at Figure 6.14. When the first system switch happens, the estimate of observer 2 is very close to the marking of the system. By switching from observer 2 to observer 1, the estimate became discontinuous and, what it is more undesirable, the estimate for the marking of p_3 becomes worse. A similar effect happens when the second system switch occurs. Another undesirable phenomenon is that the estimate of the marking of p_2 just disappears (since $\mathbf{m}[p_2]$ in unobservable for observer 1) when the estimate of observer 2 is filtered.

One way to avoid discontinuities in the resulting estimate, is to use the estimate of the observer that is going to be filtered to update the estimate of the observer that is not going to be filtered. This estimate update must be done when a system switch

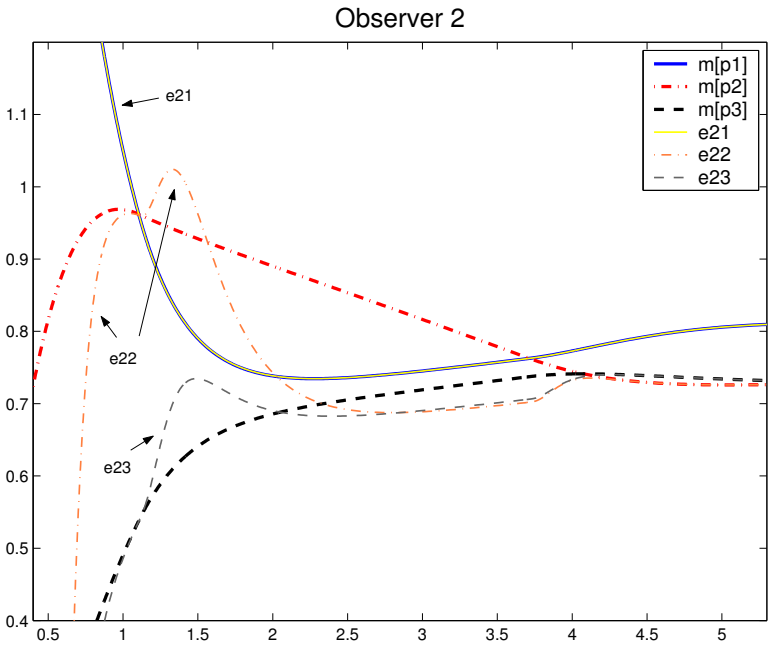


Figure 6.12: Evolution of observer 2, ($e21$, $e22$, $e23$) is the estimate for ($\mathbf{m}[p_1]$, $\mathbf{m}[p_2]$, $\mathbf{m}[p_3]$).

is detected. In order not to lose the estimate of the marking of a place when it was almost perfectly estimated (recall the case of p_2 when the first switch happened) a simulation of the system can be launched. The initial marking of this simulation is the estimate of the system just before the observability of the place is lost (in the case of the example, the estimate of observer 2 when the first switch took place). Such simulation can be seen as an estimate for those places that are not observable by the observer being considered. The simulation can only be carried out when an estimate for all the places exists and the residual is quite small. Figure 6.15 shows the evolution of the estimate of the system taking into account the following: when the first switch is detected (observer 2 becomes non-coherent) the estimate of observer 1 is initialized with the estimate of observer 2. At that point a simulation is launched to estimate the marking of p_2. When the second switch is detected (the estimate of $\mathbf{m}[p_2]$ becomes smaller than the estimate of $\mathbf{m}[p_1]$) the estimate of observer 2 is initialized with the estimate of observer 1. Notice that once the estimate is close to the system marking, it does not move away from it, even if a switch happens. Based on these ideas, Algorithm 6.12 sketches how an estimate, est, can be computed as the system evolves.

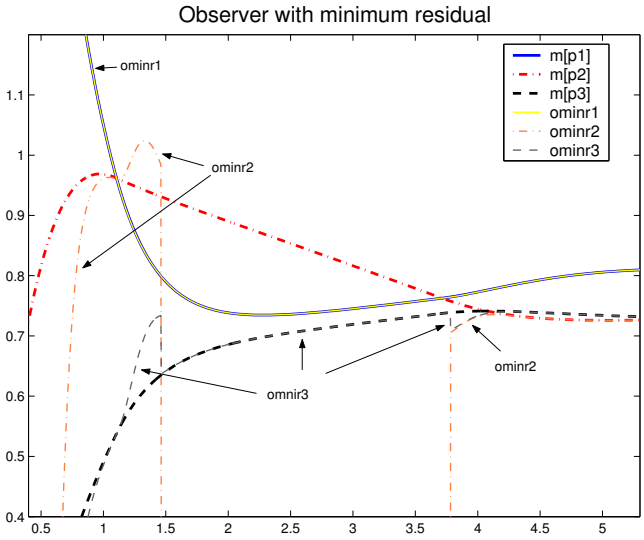


Figure 6.13: Minimum residual observer, ($ominr1$, $ominr2$, $ominr3$) is the estimate for ($\mathbf{m}[p_1]$, $\mathbf{m}[p_2]$, $\mathbf{m}[p_3]$).

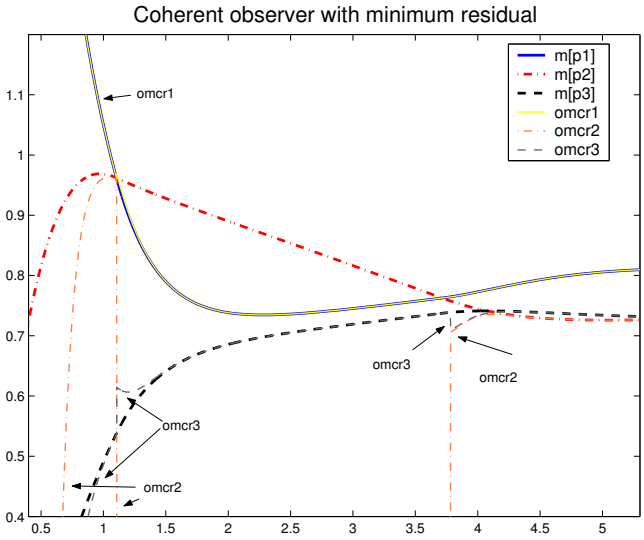


Figure 6.14: Minimum residual and coherent observer, ($omcr1$, $omcr2$, $omcr3$) is the estimate for ($\mathbf{m}[p_1]$, $\mathbf{m}[p_2]$, $\mathbf{m}[p_3]$).

Algorithm 6.12 (Observer's algorithm).

Input: ($\mathcal{N}$, $\boldsymbol{\lambda}$, $\mathcal{D}$) % Timed net and set of measured places
Output: *est* % Estimate
Begin
Launch simultaneously one observer per PT-set
Repeat
est_0 := suitable observer's estimate with minimum residual
If $est_0 \neq \emptyset$ **then** % There exists a suitable estimate
If est_0 does not estimate every place and there exists a simulation that is coherent with est_0 **then**
$est := est_0$ plus the values of the simulation that are not in est_0
Else
$est := est_0$
End_If
If a system switch is detected **then**
Update the estimates of the observers with *est*
If *est* estimates every place with small residual **then**
Create/substitute a simulation taking *est* as the initial marking
End_If
End_If
Else % No estimate is suitable
Take any observer's estimate
End_If
End_Repeat

The resulting observer can be seen as a set of switching linear observers. One of the main advantages is that the residual does not increase sharply when the PT-set of the system changes. Another interesting feature is that the use of a simulation allows one to estimate the marking of places that in some PT-sets are in principle not observable: in Figure 6.15 it can be seen that the marking of p_2 can be estimated, even when it is unobservable due to PT-set Z_1 being active.

6.6 Conclusions

Structural observability has been introduced, and studied for net systems without synchronizations (JF systems). The main advantage of considering structural observability is that it is independent of the internal speeds of transitions, $\boldsymbol{\lambda}$, i.e., a structurally observable system can be observed for any $\boldsymbol{\lambda}$. A fix point algorithm to compute the set of places that are structurally observable, has been presented. It is based on the net graph of the system, and some elementary algebraic properties.

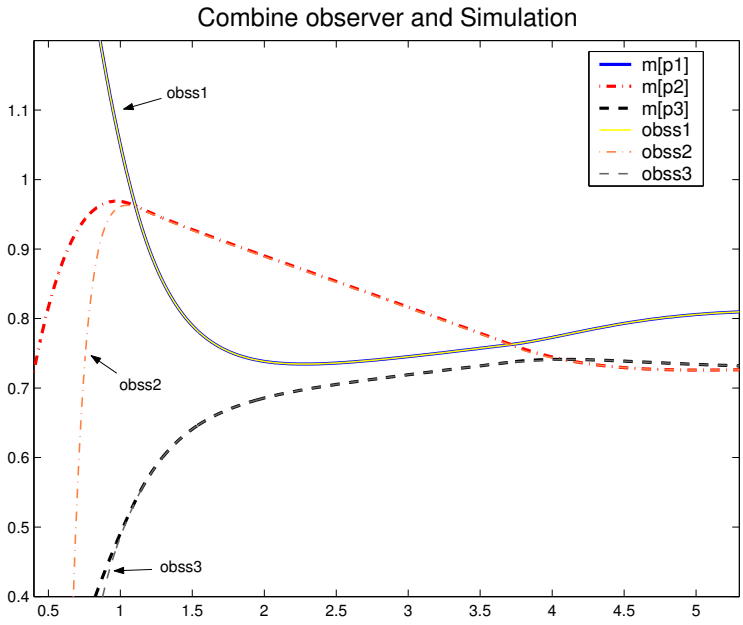


Figure 6.15: Resulting observer's estimate that makes use of a simulation, ($obss1$, $obss2$, $obss3$) is the estimate for ($\mathbf{m}[p_1]$, $\mathbf{m}[p_2]$, $\mathbf{m}[p_3]$).

A general (with synchronizations) timed continuous Petri net can be analyzed as a specific kind of piecewise linear system in which switches are triggered by internal events. By making use of some concepts of the theory for linear systems, a marking estimate can be computed for each structural PT-set. This leads to a large number of estimates. Several cases have been shown in which the estimate for a given PT-set cannot be giving a right marking estimate: The estimate is either *infeasible* or *non-coherent*. Such estimates must be "filtered" (*suspicious* estimates should also be filtered if a feasible, coherent and non-suspicious estimate exists). The PT-set ruling the evolution of the system can be identified iff only one estimate is not filtered. Given that a (deterministic) continuous Petri net system can be simulated backwards, it is enough that the system passes through an observable period in order to be able to estimate the initial marking. Thus, the conditions for observability in continuous Petri nets that have been obtained, are much less restrictive than those for general piecewise linear systems [VCS02].

With respect to the design of an observer for a timed continuous Petri net, the possibility of designing one linear (Luenberger) observer per structural PT-set has been considered. As it happens when dealing with estimates computed algebraically, the estimate yielded by a given observer may be not feasible for the linear system

for which it was designed: either the estimate generates a *non null residual* or it is *non-coherent* or *suspicious*. Based on the idea of choosing the feasible estimate with the smallest residual, a switching observer has been proposed. An interesting feature is that it launches a simulation of the system when the marking estimate is good enough. The use of such simulation allows one to improve the estimate in two ways: The estimate does not change sharply when the net PT-set switches; there is a chance of estimating the unobservable space of the PT-set driving the evolution of the system.

Chapter 7

Controllability

Summary

Optimally controlling a hybrid system is a challenging problem for which mainly continuous-time and discrete-time methods have been suggested. This chapter addresses the problem of optimal control in the framework of timed continuous Petri nets under finite servers semantics. The proposed approach consists of transforming the continuous Petri net into an equivalent hybrid system whose evolution is described by means of discrete-event steps. In particular, each step coincides with the occurrence of an event in the continuous Petri net. Thus, the number of steps required to know the behavior of the Petri net is minimum, while the accuracy is completely preserved. It is shown how to design a Mixed Integer Linear Programming problem in order to compute the optimal control solution for different performance criteria.

Introduction

Observability and controllability are dual concepts in dynamical systems theory. With respect to observability, the system is considered "untouchable" by an external agent and the goal is to extract as much information about it as possible. On the other hand, controllability is a property that applies to those systems whose state can be completely controlled by an external agent by means of input actions that explicitly modify the system behaviour. There exists a wide variety of problems related to the control of dynamical systems. The issue of optimal control is recently attracting the attention of many researches in the field of hybrid systems.

The different approaches taken to face the problem of optimal control can be roughly divided into two groups: those using continuous-time hybrid models and those using discrete-time hybrid models. Regarding continuous-time hybrid models, the main considered issues are the study of necessary trajectories to be optimal and the computation of optimal control laws by means of Hamilton-Jacobi-Bellman equations or the maximum principle. With respect to discrete-time hybrid models, a solution to optimal control problems was proposed in [BM99]. Time discretization has two important drawbacks: 1) The length of the sampling period is not easy to define. There exists a tradeoff between accuracy (short sampling period) and computational speed (long sampling period). In fact, the complexity typically grows exponentially with the number of switching variables, and these, for a given time interval, are inversely proportional to the length of the sampling period. 2) It is assumed that events can occur only at time instants that are multiple of the sampling period.

Ideally, one would like to deal with a model that requires a minimum number of steps (samples) without loss of accuracy. Consider for instance the following hybrid system expressed in continuous-time:

$$\begin{cases} \dot{x} = 2 + 2 \cdot u \text{ with } u \in [-1, 1] \text{ if } x < 8 \\ \dot{x} = 1 + 1 \cdot u \text{ with } u \in [-1, 2] \text{ if } x \geq 8 \end{cases} \tag{7.1}$$

and discretize it with a sampling period of one time unit:

$$\begin{cases} x(k+1) = x(k) + 2 + 2 \cdot u(k) \text{ with } u(k) \in [-1, 1] \text{ if } x(k) < 8 \\ x(k+1) = x(k) + 1 + 1 \cdot u(k) \text{ with } u(k) \in [-1, 2] \text{ if } x(k) \geq 8 \end{cases} \tag{7.2}$$

Let us consider the time optimal control problem of driving the system from the origin, $x(0) = 0$, to the target state $x = 14$ in minimum time. An optimal control for this problem is: $u(0) = 1$ ($x(1) = 4$), $u(1) = 1$ ($x(2) = 8$), $u(2) = 2$ ($x(3) = 11$), $u(3) = 2$ ($x(4) = 14$). Now suppose that the system has not been discretized with respect to time, but that the end of each sampling period of the model coincides with the occurrence of an event, i.e., a switch in the dynamics of the system. The time

elapsed between events is now a real variable q. Such a model may be classified as *event-driven*. If that model could be created the time optimal control to reach $x = 14$ would be: $u(0) = 1$ during $q(0) = 2$ time units ($x(1) = 8$), $u(1) = 2$ during $q(1) = 2$ time units ($x(2) = 14$). That is, only two steps are necessary. In fact, given the linearity of the system, its whole evolution can be derived from the state at the event instants, and therefore steps 1 and 3 of the discrete-time model could be avoided.

The idea of optimal control via event-driven models is a relatively novel approach for controlling hybrid systems (see for example [SvdB01]). In this chapter, such an approach is presented and applied to the model of continuous Petri nets (PN) [JBRS04].

The rest of the chapter is organized as follows: In Section 7.1 it will be shown how control actions can be introduced into a timed continuous Petri net model in order to control it. Section 7.2 is concerned with the transformation of a continuous Petri net into a novel event-based Mixed Logical Dynamical System (eMLD) [BM99], a class of hybrid models that is very suitable for optimization purposes. The most important feature of this transformation is that the obtained eMLD system switches its continuous-time dynamics if and only if an event occurs in the original continuous Petri net. Section 7.3 shows how optimal control problems can be solved by using Mixed Integer Linear Programming techniques.

7.1 Controlled Petri net systems

Control actions can be introduced into the model in order to modify the *autonomous* timed evolution of the system. In continuous PNs these actions are applied to the transitions of the net. A transition t is *controllable* when its flow can be slowed down in a quantity that depends on the input, $\mathbf{u}[t]$, applied to it. The value $\mathbf{u}[t]$ is positive and upper limited by $\boldsymbol{\lambda}[t]$. An action $\mathbf{u}[t]$ on the transition t can be seen as if the valve associated to t was closed in an amount $\mathbf{u}[t]$. The way of computing $\mathbf{f}$ is analogous to the one shown in Subsection 1.3.1, being now the maximum flow allowed by t, $\boldsymbol{\lambda}[t] - \mathbf{u}[t]$. Hence, if transition t is strongly enabled then $\mathbf{f}[t] = \boldsymbol{\lambda}[t] - \mathbf{u}[t]$. If t is weakly enabled $\mathbf{f}[t]$ will be computed considering $\boldsymbol{\lambda}[t] - \mathbf{u}[t]$ the upper bound for the flow of t. If t is neither strongly nor weakly enabled $\mathbf{f}[t] = 0$.

To show how input actions modify the evolution of a system, let us apply the input action $\mathbf{u}[t_1] = 0.5$ to the system in Figure 7.1(a). Since, transition t_1 is initially strongly enabled, its flow will be $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1] = 1.5$. After two time units p_1 becomes empty. Hence, the maximum flow allowed by t_1 is the input flow coming to p_1, that is 1. Now, every input action on t_1 ranging from 0 to 1 has no effect on the system evolution. However, if $\mathbf{u}[t_1]$ is greater that 1, the flow of t_1 will be slowed down. For example, if $\mathbf{u}[t_1] = 1.5$, the flow of t_1 will be $\mathbf{f}[t_1] = 0.5$, and consequently p_1 will start to fill. The analytic expression for $\mathbf{f}[t_1]$ with arbitrary $\boldsymbol{\lambda}[t_1]$ and $\boldsymbol{\lambda}[t_2]$ when p_1 is empty is $\mathbf{f}[t_1] = \min(\boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1], \boldsymbol{\lambda}[t_2] - \mathbf{u}[t_2])$.

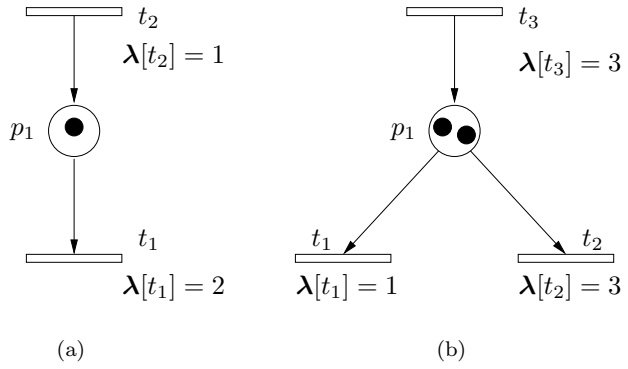


Figure 7.1: The input actions applied to the transitions determine their flow.

Similarly, for the system in Figure 7.1(b), if $\mathbf{m}[p_1] > 0$ then $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1]$ and $\mathbf{f}[t_2] = \boldsymbol{\lambda}[t_2] - \mathbf{u}[t_2]$. If $\mathbf{m}[p_1] = 0$ then

$$\mathbf{f}[t_1] = \min((\boldsymbol{\lambda}[t_3] - \mathbf{u}[t_3])/2 + \max(0, (\boldsymbol{\lambda}[t_3] - \mathbf{u}[t_3])/2 - (\boldsymbol{\lambda}[t_2] - \mathbf{u}[t_2])), \boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1])$$

and

$$\mathbf{f}[t_2] = \min((\boldsymbol{\lambda}[t_3] - \mathbf{u}[t_3])/2 + \max(0, (\boldsymbol{\lambda}[t_3] - \mathbf{u}[t_3])/2 - (\boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1])), \boldsymbol{\lambda}[t_2] - \mathbf{u}[t_2]).$$

7.2 Modelling continuous Petri nets as event-driven MLD systems

7.2.1 Mixed Logical Dynamical systems

Mixed logical dynamical (MLD) systems [BM99] are computationally oriented representations of hybrid systems. They consist of a set of linear equalities and inequalities involving both real and Boolean $(0, 1)$ variables. An MLD system is described by the following relations:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B_1\mathbf{u}(k) + B_2\delta(k) + B_3\mathbf{z}(k) + B_5 \tag{7.3}$$

$$\mathbf{y}(k) = C\mathbf{x}(k) + D_1\mathbf{u}(k) + D_2\delta(k) + D_3\mathbf{z}(k) + D_5 \tag{7.4}$$

$$E_2\delta(k) + E_3\mathbf{z}(k) \leq E_1\mathbf{u}(k) + E_4\mathbf{x}(k) + E_5 \tag{7.5}$$

where $\mathbf{x} \in \mathbb{R}^{n_r} \times \{0,1\}^{n_b}$ is a vector of continuous and binary states, $\mathbf{u} \in \mathbb{R}^{m_r} \times \{0,1\}^{m_b}$ are the inputs, $\mathbf{y} \in \mathbb{R}^{p_r} \times \{0,1\}^{p_b}$ are the outputs, $\delta \in \{0,1\}^{r_b}$,

$\mathbf{z} \in \mathbb{R}^{r_r}$ represent auxiliary binary and continuous variables, respectively, and $A, B_1, B_2, B_3, C, D_1, D_2, D_3, E_1, E_2, E_3, E_4, E_5$ are matrices of suitable dimensions. Given the current state $\mathbf{x}(k)$ and input $\mathbf{u}(k)$, the evolution of (7.3)-(7.5) is determined by solving $\delta(k)$ and $\mathbf{z}(k)$ from (7.5) and then updating $\mathbf{x}(k+1)$ and $\mathbf{y}(k)$ from (7.3) and (7.4). It is assumed that the system (7.3)-(7.5) is completely *well-posed* [BM99], which means that for all $\mathbf{x}(k)$, $\mathbf{u}(k)$ within a given bounded set the variables $\delta(k)$, $\mathbf{z}(k)$ are defined by (7.5) in a unique way.

Several conversion laws exist that allow one to transform logic relations into mixed-integer inequalities. For example the threshold condition

$$[\delta = 1] \leftrightarrow [a \cdot x + b \cdot u + c \geq 0] \tag{7.6}$$

where $\delta \in \{0, 1\}$, $a, b, c, x, u \in \mathbb{R}$ can be equivalently expressed as:

$$\begin{cases} a \cdot x + b \cdot u + c < M \cdot \delta \\ a \cdot x + b \cdot u + c \geq m \cdot (1 - \delta) \end{cases} \tag{7.7}$$

where M, m are upper and lower bounds, respectively, on $a \cdot x + b \cdot u + c$. In a similar way, the semantics of the common relation

$$\text{IF } \delta \text{ THEN } z = a_1 \cdot x + b_1 \cdot u + c_1 \text{ ELSE } z = a_2 \cdot x + b_2 \cdot u + c_2 \tag{7.8}$$

which links Boolean to continuous variables, can be expressed with the following set of inequalities:

$$\begin{cases} (m_2 - M_1) \cdot \delta + z \leq a_2 \cdot x + b_2 \cdot u + c_2 \\ (m_1 - M_2) \cdot \delta - z \leq -a_2 \cdot x - b_2 \cdot u - c_2 \\ (m_1 - M_2) \cdot (1 - \delta) + z \leq a_1 \cdot x + b_1 \cdot u + c_1 \\ (m_2 - M_1) \cdot (1 - \delta) - z \leq -a_1 \cdot x - b_1 \cdot u - c_1 \end{cases} \tag{7.9}$$

where $\delta \in \{0, 1\}$, $a_1, b_1, c_1, a_2, b_2, c_2, x, u, z \in \mathbb{R}$ and M_i, m_i are upper and lower bounds on $a_i \cdot x + b_i \cdot u + c_i$, $i = 1, 2$. Further details on these and other conversions can be seen for example in [Mig02]. These transformations allow MLD systems to model a great variety of hybrid systems.

Usually, in an MLD system k represents a time-step counter, and the length of the time step is constant. It is common that the shorter the time step the greater the accuracy of the model. Clearly, the main drawback of having a very short time step is that many steps may be required to study the evolution of the system during a given time interval. Assuring good accuracy while minimizing the number of steps is a good criterion to choose the length of the time step.

7.2.2 Continuous Petri nets as event-driven MLD systems

It will be shown how to use the MLD transformation machinery (see e.g. [TB04]) in order to describe the behavior of a *non-controlled* continuous PN. In a non-controlled system no input actions are considered, and therefore, the value of $\mathbf{f}$ is constant between events and depends only on the IB - state of the net. By treating each step k as the occurrence of an event in the continuous PN rather than the elapse of a sampling period, the continuous PN will be transformed into an event-based mixed logical dynamical (eMLD) system, i.e., an MLD such that after each step a place becomes empty. Observe that this approach has two interesting advantages:

- Event-discretization does not imply loss of accuracy: The marking evolution of a continuous PN is linear between events, and so it can be determined from the marking of the net at the event instants.
- The number of steps is minimized: A step happens only when it is really required (an event happens).

Clearly, this implies that the length of the period cannot be constant but depends on the event instants, therefore making the eMLD system one is dealing with a truly *event-driven* system, rather than a time-drive one. The time period will be a real variable of the eMLD system expressing the length of the interval between two events.

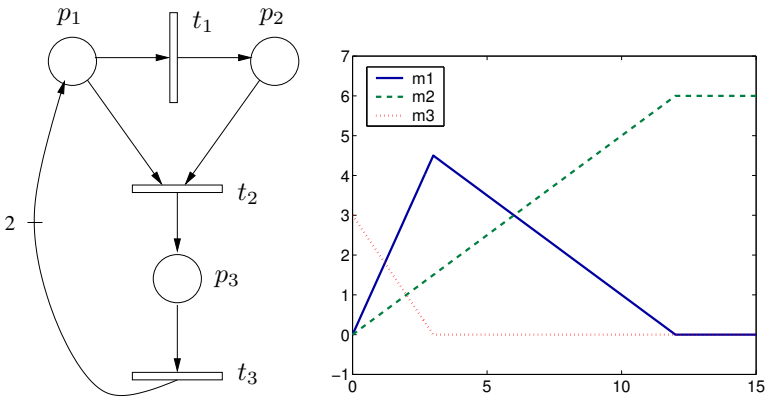
The steps to convert a continuous PN into the aforementioned eMLD system are the following:

1. *Identify the potential IB - states of the continuous PN.* That is, the potential dynamics that may rule the evolution of the system. For example, the system in Figure 7.4 has four potential IB - states: a) $\mathbf{m}[p_1] > 0$, $\mathbf{m}[p_2] > 0$, b) $\mathbf{m}[p_1] > 0$, $\mathbf{m}[p_2] = 0$ c) $\mathbf{m}[p_1] = 0$, $\mathbf{m}[p_2] > 0$ d) $\mathbf{m}[p_1] = 0$, $\mathbf{m}[p_2] = 0$.
2. *Describe the behavior of the PN under each IB - state.* This is equivalent to define the flow of the transitions under each IB - state.
3. *Define the evolution of the marking.* As seen in Subsection 1.3.1, the derivative of the marking is given by the incidence matrix multiplied by the flows of the transitions. This has to be included in the set of equations of the eMLD system. For the system in Figure 7.1(b), the equation defining the evolution of the marking is: $\mathbf{m}(k+1) = \mathbf{m}(k) + q \cdot (\mathbf{f}[t_3] - \mathbf{f}[t_1] - \mathbf{f}[t_2])$, where q is the real variable storing the time elapsed between events k and $k+1$.
4. *Force that at least one place becomes empty at the occurrence of the next event.* To achieve this, a Boolean variable per place can be defined. The Boolean variable becomes true iff the place at event k is marked and becomes empty

after q time units. By means of equation (7.5) it is easy to force that the sum of these Boolean variables is positive, i.e., at least one place becomes empty.

The task of transforming an event-driven model describing a continuous PN into an eMLD system in the form of (7.3) is greatly eased by HYSDEL [TB04] (HYbrid System DEscription Language). Basically, HYSDEL allows one to describe a hybrid system in textual form and generates the matrices of the equivalent MLD form.

Let us consider the system in Figure 7.2(a) with $\boldsymbol{\lambda} = (1.5\ 1\ 2)$ and $\mathbf{m_0} = (0\ 0\ 3)$. The system is transformed into eMLD form following the tasks described above. Only two steps are necessary to reach the steady state marking. The length of the first two intervals is respectively $q(1) = 3$ and $q(2) = 9$. The evolution of the system is depicted in Figure 7.2(b). After the second step, i.e., event, p_1 and p_3 become empty and the system dies, that is, the flow of every transition is zero in the steady state.



(a) (b)

Figure 7.2: (a) A continuous PN system. (b) The associated eMLD requires two steps to reach the steady state.

7.3 Optimal control using Mixed Integer Linear Programming

7.3.1 Obtaining a Mixed Integer Linear Programming

The eMLD system obtained in the previous section must be slightly modified in order to take into account the effect of the control actions in the marking evolution. As

explained in Subsection 7.1, when a transition t is controllable its flow, $\mathbf{f}[t]$, depends on the input action $\mathbf{u}[t]$ applied to it. Let us assume that the control actions are constant between events. Notice that this constraint is not very strong since the effect of a non constant $\mathbf{u}[t]$ is equivalent to the effect of a constant $\mathbf{u}[t]$ with the same integral.

Let us consider again the system in Figure 7.1(a) with $\mathbf{m_0}[p_1] = 1$. Suppose that transition t_1 is controllable. Now, the flow of t_1 is $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1]$ and the marking evolution of p_1 from event k to $k+1$ is $\mathbf{m}(k+1) = \mathbf{m}(k) + q \cdot (\mathbf{f}[t_2] - \mathbf{f}[t_1]) = \mathbf{m}(k) + q \cdot (\boldsymbol{\lambda}[t_2] - (\boldsymbol{\lambda}[t_1] - \mathbf{u}[t_1]))$. Thus, the equation defining $\mathbf{m}(k+1)$ becomes nonlinear since both q and $\mathbf{u}[t_1]$ are real variables. Such an equation cannot be included directly in an eMLD system. A way of overcoming this situation is to define a new real variable $\mathbf{h}[t_1]$ as follows: $\mathbf{h}[t_1] = q \cdot \mathbf{u}[t_1]$. Hence, $\mathbf{f}[t_1] \cdot q = q \cdot \boldsymbol{\lambda}[t_1] - \mathbf{h}[t_1]$ and no product of real variables is required to compute $\mathbf{m}(k+1)$. That is, there does not exist an explicit input variable but two real variables q and $\mathbf{h}[t_1]$ from which the value of $\mathbf{u}[t_1]$ can be obtained.

The input action on a controllable transition, $\mathbf{u}[t]$, must be bounded by 0 and $\boldsymbol{\lambda}[t]$. Equation (7.5) can be used to express these bounds. For example, for the system in Figure 7.1(a) the equation $0 \leq \mathbf{h}[t_1] \leq q \cdot \boldsymbol{\lambda}[t_1]$ must be added.

The following step after introducing the input actions into the eMLD system is to define the function to be optimized. Let us focus on linear optimization functions. The optimization variables that can be included in that function are: q, $\mathbf{h}$, $\mathbf{m}$ or any other Boolean or real variable that can be linearly originated from these ones. Furthermore, if desired, the eMLD formalism allows one to add constraints on the mentioned variables by means of Equation (7.5). This all leads to a multivariable optimal control problem that can be expressed as:

$$\begin{aligned} &\min \mathbf{f} \cdot [\mathbf{m}(0), ., \mathbf{m}(N), \mathbf{h}(0), ., \mathbf{h}(N), \mathbf{z}(0), ., \mathbf{z}(N), \delta(0), ., \delta(N)] \\ &s.t. \begin{cases} \mathbf{m}(k+1) = A\mathbf{m}(k) + B_1\mathbf{h}(k) + B_2\delta(k) + B_3\mathbf{z}(k) + B_5 \\ E_2\delta(k) + E_3\mathbf{z}(k) \leq E_1\mathbf{u}(k) + E_4\mathbf{x}(k) + E_5 \end{cases} \end{aligned} \tag{7.10}$$

Note that the duration of each period is stored in the real variable q, that is part of the vector of real auxiliary variables $\mathbf{z}$. This control problem is defined for a horizon of N steps. It can be seen as a Mixed Integer Linear Programming (MILP) problem where the integer variables can only be 0 or 1. See [Flo95] for a review of methods to solve MILP problems. The solution of the programming problem contains the input actions that have to be applied to the continuous PN in order to optimally control it.

Figure 7.3 shows the steps that have been followed to obtain an MILP problem from the initial continuous PN. In contrast to the eMLDs described in Subsection 7.2.2 the eMLDs in this Section do not represent the free evolution of the system but its controlled evolution. The controlled system will evolve in such a way that the

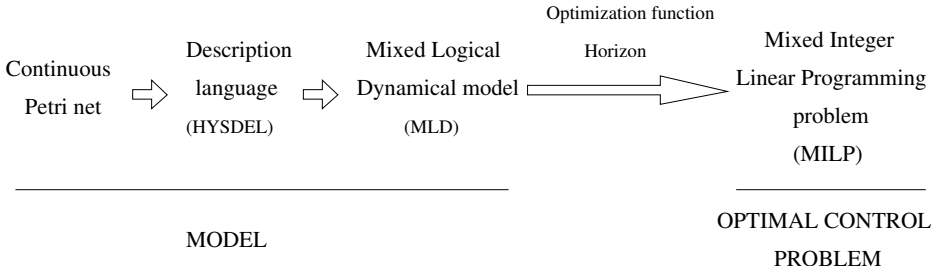


Figure 7.3: Obtaining an MILP problem from a continuous PN.

optimization function is minimized/maximized. Events will happen without being forced so that the system behaves optimally. Therefore it is not necessary to force the occurrence of events when writing the eMLDs, i.e., step 4 in Subsection 7.2.2 can be skipped.

7.3.2 Optimality Criteria

The optimization examples of this Subsection show how different kinds of optimal control problems are solved by means of the explained event-driven approach. The control problems have to do with reaching a target marking in minimum time, i.e., time optimal control, maximizing the steady state throughput and maximizing an optimization function in which several different parameters are involved.

Time optimal control

Consider the system in Figure 7.4 where $\mathbf{m_0} = (4\ 2)$, $\boldsymbol{\lambda} = (2\ 4\ 3\ 2)$ and the controllable transitions are t_1 and t_2. Let us consider the optimal control problem of reaching the target marking $\mathbf{m} = (0\ 5)$ in minimum time. Hence the function to minimize in (7.10) is $\sum_{i=0}^{N} q(i)$. It is obtained $\mathbf{u}[t_1](0) = 0$, $\mathbf{u}[t_2](0) = 2.4286$ and the duration of the step is $q(0) = 7$. The target marking is reached in one step. If it is desired to know the steady state control at the target marking, it is only necessary to force the marking in the last period to be constant. The steady state control obtained is $\mathbf{u}[t_1] = 1$, $\mathbf{u}[t_2] = 2$.

The inclusion of auxiliary Boolean variables in (7.10) allows one to define more elaborated optimization functions that may be useful in real situations. For example, for the system in Figure 7.4, a Boolean variable per place can be defined in the following way: The Boolean variable associated to p_1 (respectively p_2) is true iff the target marking of p_1 (respectively p_2) has been reached. By using these Boolean variables, it is possible to assign different priorities to different places, e.g., to favor a certain place. Assume that a penalty of 3 (respectively 1) units is obtained per

time unit elapsed without reaching $\mathbf{m}[p_1] = 0$ (respectively $\mathbf{m}[p_2] = 5$). An optimal control problem that minimizes the sum of penalties yields the following control: $\mathbf{u}[t_1](0) = 0$, $\mathbf{u}[t_2](0) = 1$ during the first step of $q(0) = 2$ units reaching $\mathbf{m}(1) = (0\ 0)$ and $\mathbf{u}[t_1](1) = 0$, $\mathbf{u}[t_2](1) = 3$ during the second step of $q(1) = 5$ units reaching $\mathbf{m}(2) = (0\ 5)$.

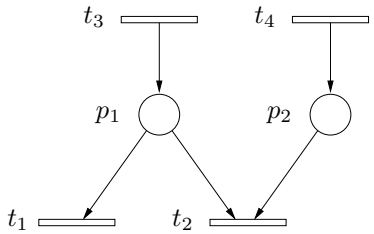


Figure 7.4: A continuous PN with controllable transitions t_1 and t_2.

Consider the system in Figure 7.2(a) with $\boldsymbol{\lambda} = (1.5\ 1\ 2)$, $\mathbf{m_0} = (6\ 0\ 0)$ with t_1 and t_3 as controllable transitions. Let us give a penalty of 1 unit per time unit per place whose target marking has not been reached. The optimal control to reach $\mathbf{m} = (0\ 4\ 1)$ that minimizes the penalty is: $\mathbf{u}[t_1](0) = 0$, $\mathbf{u}[t_3](0) = 2$ during the first step of $q(0) = 1$ units reaching $\mathbf{m}(1) = (3.5\ 0.5\ 1)$ and $\mathbf{u}[t_1](1) = 0$, $\mathbf{u}[t_3](1) = 1$ during the second step of $q(1) = 7$ units reaching $\mathbf{m}(2) = (0\ 4\ 1)$.

Maximization of the steady state throughput

If no control is applied to the system in Figure 7.2(a) with $\boldsymbol{\lambda} = (1.5\ 1\ 2)$, it reaches a steady state with null throughput, i.e., flow, for any initial marking. An interesting control problem for such non live systems and for many manufacturing systems is to maximize the throughput in the steady state. Consider that system with $\mathbf{m_0} = (6\ 0\ 0)$ and t_3 as the only controllable transition. Let us define an optimal control problem that maximizes the throughput of t_1 in the last period. The duration of the last period is required to be 10 time units. During the last period the marking must be kept constant, i.e., it is a steady state marking. Notice that the system has a unique T-semiflow (repetitive sequence) and therefore maximizing the throughput of t_1 in the steady state is equivalent to maximizing the throughput of any of the three transitions. The obtained control is: $\mathbf{u}[t_3](0) = 2$ during $q(0) = 2.4$ reaching $\mathbf{m}(1) = (0\ 1.2\ 2.4)$ and $\mathbf{u}[t_3](1) = 1$ for the steady state ($q(1) = 10$). At that marking the flow of the transitions is 1.

Optimization based on several parameters

The PN in Figure 7.5 represents a simple manufacturing system with two lines, (t_1, t_3) and (t_2, t_4), and a shared resource p_5. Let t_1 and t_4 be the controllable transitions, $\mathbf{m_0} = (4\ 3\ 1\ 1\ 2)$ and $\boldsymbol{\lambda} = (2\ 3\ 5\ 2)$. Suppose that one is interested in reaching a steady state marking in which $\mathbf{m}[p_1] + \mathbf{m}[p_4]$ is maximum. Besides, a steady state in which the flow of the transitions is as high as possible is desirable. To do this, an optimization function including markings and flows can be defined. Weights can be assigned to the markings and flows to highlight the importance of each parameter in the optimization function. For example, an optimization function that gives the same weight to the marking and the flows is: $\mathbf{m}[p_1](N) + \mathbf{m}[p_4](N) + \mathbf{f}[t_1](N) + \mathbf{f}[t_2](N)$. Due to the existing T-semiflows, in the steady state t_3 (t_4) will have the same flow than t_1 (t_2). Therefore, in the optimization function it is not necessary to include neither $\mathbf{f}[t_3]$ nor $\mathbf{f}[t_4]$.

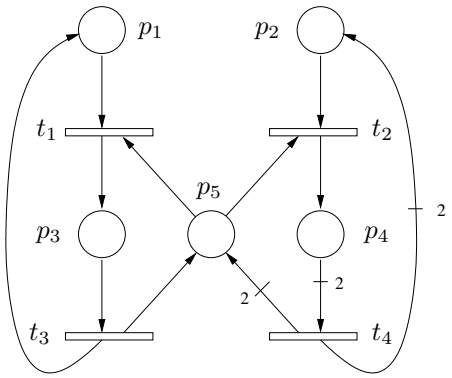


Figure 7.5: A small manufacturing system.

Furthermore, it would be nice to reach the steady state marking as soon as possible. For this control problem, an adequate optimization function to be maximized is $k \cdot (\mathbf{m}[p_1](N) + \mathbf{m}[p_4](N)) + k' \cdot (\mathbf{f}[t_1](N) + \mathbf{f}[t_2](N)) - \sum_{i=0}^{N} q(i)$ with k, k' big enough to ensure that the marking of places and flows have priority (in this case it is enough $k, k' \geq 10$). The obtained control with $k = k' = 10$ is: $\mathbf{u}[t_1](0) = 2$, $\mathbf{u}[t_4](0) = 2$ with $q(0) = 0.2$ reaching $\mathbf{m}(1) = (5\ 2.4\ 0\ 1.6\ 2.4)$, $\mathbf{u}[t_1](1) = 0$, $\mathbf{u}[t_4](1) = 2$ with $q(1) = 0.8$ reaching $\mathbf{m}(2) = (5\ 0\ 0\ 4\ 0)$ that is a steady state marking with the inputs $\mathbf{u}[t_1](2) = 0$, $\mathbf{u}[t_4](2) = 0.5$.

The main drawback of solving an MILP to obtain an optimal control law is that the time complexity is exponential with respect to the number of boolean variables. This way, even the solution for not very large systems requires a substantial amount of time. The control problems presented in this Section were solved by a PC Pentium IV 2.6 Ghz using the GLPK (GNU Linear Programming Kit) package solver running

under Matlab 6.5. The optimal solutions were always found in less than 3 minutes.

7.4 Conclusions

This chapter has studied the problem of optimally controlling a continuous Petri net. Input actions have been explicitly introduced into the model and their effect in the evolution of the system has been defined. The main concern is to overcome the drawbacks that appear when considering discrete-time models. That is, loss of accuracy when using a too long period and high computational load when using a too short period.

The main contribution is to obtain a hybrid model with an *event-based* discretization. That is, each step of the system coincides with the occurrence of an event. Given that the evolution of continuous Petri nets is linear between events, the whole trajectory of the system can be reconstructed from the marking at event instants. This way, the number of required steps is minimized and the accuracy preserved. To obtain such a hybrid model, continuous Petri nets are transformed into event-based mixed logical dynamical models with some particular features. The use of the latter models and a linear optimization function allows one to find the input actions that optimally control the original continuous Petri net by solving a mixed integer linear programming problem.

Chapter 8

Cases of Study

Summary

Three cases of study are reported in this chapter. First, a manufacturing system representing a table factory is considered. The second case refers to an assembly line with kanban strategy. For these systems, even for small initial markings, the size of the reachability set is very large if the Petri net is considered as discrete. In this thesis two servers semantics have been studied: Finite servers and infinite servers semantics. For these two first cases, infinite servers semantics is considered; The first case also deals with finite servers semantics. Many of the results obtained in this thesis are used to study the properties of those systems as continuous. The main goal of the third case is to model a car traffic system. This case, introducing the idea of "realism", provides a third time interpretation as a "variant" of infinite servers semantics. This indicates that servers semantics is still a topic that requires much research effort, being the case now that for this "new" time interpretation essentially simulation can be used. The addition of discrete elements (traffic lights) to the model results in a hybrid Petri net.

8.1 A Manufacturing System

The Petri net system in Figure 8.1 represents a manufacturing system that produces tables [Ter94, RS00]. The choice of such a simple case study is deliberate, the simplicity of the model allows a clear presentation of the analysis techniques without getting confused in non relevant details. The techniques presented here can be directly applied on more complex manufacturing systems.

The system is composed of the following items: Two different machines (t_1 and t_2) to make table-legs, a new fast one (t_1) which produces two legs at a time, and the old one (t_2), which makes legs one by one; A machine (t_3) to produce table boards; A machine (t_5) to assemble a 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, boards are stored in p_6, and p_7 is devoted to the storage of unpainted tables. The rest of the 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. Due to some commercial considerations, it is desired 50% of the tables to be assembled in the factory and 50% to be brought from the other factory (this order is represented by equal weights of the arcs going from t_6 to p_3 and p_4). It is also required that 75% of the legs are produced by the new machine and 25% by the old one (this is modelled by the arc weights going from t_6 to p_1 and p_2).

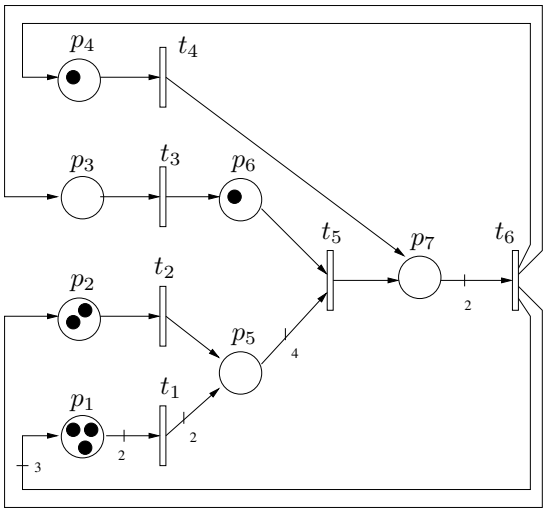


Figure 8.1: Petri net modelling a table factory.

The size of the reachability set if the system in Figure 8.1 is considered as discrete

is 54. If the initial marking is scaled the size of the reachability set increases exponentially (see Figure 1.1). Even for this small manufacturing example, analysis techniques based on an exhaustive exploration of the state space are computationally prohibitive if a large initial marking is considered. The remainder of this section studies the fluidified model of the manufacturing system, aiming to establish a connection between the properties of the continuous model and those of a highly populated discrete one.

General considerations

The Petri net that models the manufacturing system is conservative and consistent, i.e., every place is covered by a P-semiflow and every transition is covered by a T-semiflow. The minimal P-semiflows of the net are (1 1 0 4 1 0 4) and (0 0 1 1 0 1 1). There exists only one minimal T-semiflow, (3 2 2 2 2 2), thus, the net is mono-T-semiflow (Subsection 1.1.2). Therefore, the firing proportions of the transitions after any infinitely long firing sequence are determined by the T-semiflow. Furthermore, if time is introduced to the model, in the steady state the flow through transitions under any time interpretation has to be proportional to the T-semiflow. The firing rates of the transitions, both for infinite and finite servers semantics, are $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_2] = \boldsymbol{\lambda}[t_3] = \boldsymbol{\lambda}[t_4] = \boldsymbol{\lambda}[t_5] = \boldsymbol{\lambda}[t_6] = 1$.

The system has one synchronization at transition t_5 (machine assembling boards and legs) involving places p_5 and p_6. Hence, according to Subsection 1.3.2, if infinite servers semantics is considered the system has two structural PT-sets: $W_1 = \{(p_1,t_1),(p_1,t_2),(p_3,t_3),(p_4,t_4),(p_5,t_5),(p_7,t_6)\}$ that will be active if $\mathbf{m}[p_5]/4 \leq \mathbf{m}[p_6]$ and $W_2 = \{(p_1,t_1),(p_1,t_2),(p_3,t_3),(p_4,t_4),(p_6,t_5),(p_7,t_6)\}$ that will be active if $\mathbf{m}[p_6] \leq \mathbf{m}[p_5]/4$. If W_1 is active then the system marking evolves according to:

$$\Sigma_1 : \quad \dot{\mathbf{m}} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1.5 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -0.25 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0.25 & 0 & -1 \end{pmatrix} \cdot \mathbf{m} \tag{8.1}$$

Otherwise, if W_2 is active then the evolution of the marking is driven by:

$$\Sigma_2: \quad \dot{\mathbf{m}} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1.5 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{pmatrix} \cdot \mathbf{m} \tag{8.2}$$

If $\mathbf{m}[p_6] = \mathbf{m}[p_5]/4$ any of the equations can be taken since both yield the same value. Notice that if W_1 is active, place p_6 is time implicit and does not play any role in the system evolution (the sixth column of the matrix in Σ_1 is null). The same happens to p_5 if W_2 is active.

Figure 8.2 depicts the marking evolution under infinite servers semantics. Initially, the PT-set W_1 is active. At time $\tau = 1.61$ a commutation occurs ($\mathbf{m}[p_6]$ becomes less than $\mathbf{m}[p_5]/4$) and W_2 becomes active.

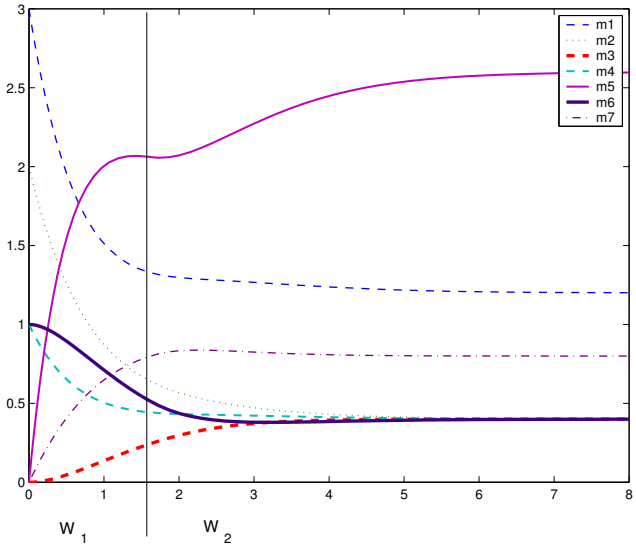


Figure 8.2: Evolution of the manufacturing system under infinite servers semantics.

Reachability

The net is consistent and every transition can be fired, thus, by Corollary 2.18 it holds lim-RS($\mathcal{N}, \mathbf{m_0}$) = δ-RS($\mathcal{N}, \mathbf{m_0}$) = LRS($\mathcal{N}, \mathbf{m_0}$), i.e., there are no spurious solutions of the state equation if lim-reachability or δ-reachability are considered.

The set of reachable markings with a finite firing sequence is however not equal to the linearized reachability set. The net system has several traps that are initially marked and cannot be emptied with a finite firing sequence. For example, the traps $\Theta_1 = \{p_4, p_7\}$ and $\Theta_2 = \{p_2, p_5, p_7\}$ cannot be emptied with a finite firing sequence, nevertheless they can be emptied with infinite firing sequences whose firing count vectors are $\boldsymbol{\sigma}_1$ = (0 0 0 2 0 1) and $\boldsymbol{\sigma}_2$ = (0.5 3 0 1 1 1). Given that the system is consistent, every transition is fireable and every trap is initially marked, the conditions in Theorem 2.12 for a marking to be reached reduce to being a solution of the state equation that does not contain empty traps. This result is obtained by reasoning similarly as in Corollary 2.16. In other words, the set of reachable markings with a finite firing sequence is equal to the linearized reachability set minus those markings that contain empty traps.

Liveness

Since the net is mono-T-semiflow deadlock-freeness is equivalent to liveness. The only markings at which no transition can be fired, i.e., deadlock markings, take place either if $\mathbf{m}[p_1] = \mathbf{m}[p_2] = \mathbf{m}[p_3] = \mathbf{m}[p_4] = \mathbf{m}[p_5] = \mathbf{m}[p_7] = 0$ or $\mathbf{m}[p_1] = \mathbf{m}[p_2] = \mathbf{m}[p_3] = \mathbf{m}[p_4] = \mathbf{m}[p_6] = \mathbf{m}[p_7] = 0$. Both situations would imply that a P-semiflow has become empty, thus, those deadlock markings are not solution of the state equation (Proposition 5.1) and cannot be reached under any reachability concept. The untimed system is therefore live, lim-live and δ-live as continuous (hence, it is also structurally live as discrete [Rec98]).

Since liveness of the untimed system is a sufficient condition for liveness of the timed system (Section 4.2), it can be asserted that the timed system is live under any time interpretation. Obviously, it verifies the condition stated in Theorem 4.13 for robust-liveness.

Performance Evaluation

In the steady state, the set of places constraining the firing of the transitions (T-coverture) is either $\{p_1, p_2, p_3, p_4, p_5, p_7\}$ or $\{p_1, p_2, p_3, p_4, p_6, p_7\}$. Both T-covertures contain a P-semiflow. Thus, according to Corollary 5.6 the exact throughput of the system under infinite servers semantics can be computed in polynomial time with the LPP (5.19). The solution of that LPP yields $\mathbf{f}[t_6] = 0.4$, the bottleneck P-semiflow associated to the solution is (0 0 1 1 0 1 1).

In order to check whether the fluidified model faithfully represents the discrete one with large populations, it would be interesting to compare the throughput of the system seen as discrete and as continuous. Table 8.1 shows how the throughput of the markovian discrete system changes as the initial marking is scaled by increasing quantities (k). The last column reports on the relative error of the normalized throughput of the discrete system with respect to the throughput of the continuous system $\mathbf{f}[t_6] = 0.4$. Notice how this relative error decreases as the scale constant is increased.

k	Reachable markings	$\chi[t_6]$	$\chi[t_6]/k$	Relative error
1	54	0.2522	0.2522	36.95%
2	1685	0.6493	0.3246	18.85%
3	10354	1.0567	0.3522	11.95%
4	37722	1.4623	0.3656	8.60%
5	103914	1.8671	0.3734	6.65%

Table 8.1: Throughput of the discrete system as the initial marking is scaled, normalized throughput and relative error with respect to the throughput of the continuous system.

Observability

Let us assume that the places p_1, p_3, p_4 and p_7 of the system cannot be measured, but there are sensors that allow one to know the marking of places p_2, p_5 and p_6. Following the notation in Section 6.1 the matrix $\mathbf{S}$ is (0 1 0 0 0 0 0; 0 0 0 0 1 0 0; 0 0 0 0 0 1 0).

Equation 6.1 can be used to compute one estimate for the PT-set W_1 and another estimate for the PT-set W_2. The observability matrix, ϑ, has full rank for both PT-sets W_1 and W_2. Thus, the non observable subspace is empty for both PT-sets. Assuming that the system is not initially at a steady state marking in which $\mathbf{m}[p_6] = \mathbf{m}[p_5]/4$ (both PT-sets coincide), only one of the estimates is feasible and coherent. Hence (Proposition 6.9), the initial marking can be computed.

The use of more sensors in the system does not always result in an enlargement of the observable subspace. For instance, if p_1, p_4, p_5 and p_7 are measured instead of p_2, p_5 and p_6 the observability matrix for W_2 has full rank but its rank is only 5 for W_1. More precisely, the marking of places p_3 and p_6 cannot be computed if W_1 is active. This way, it turns out to be more important to choose an appropriate location for the sensors than to increase its number without a good criterion. In this case, the initial PT-set is W_1 and therefore the marking of places p_3 and p_6 cannot in principle be computed. Fortunately the system marking evolves to W_2 at which it

can be completely observed. Once the complete marking at a given time instant is obtained, a backward simulation of the system makes possible to compute the initial marking of the system (Proposition 6.10).

Control

The transformation of the net system into an event-based Mixed Logical Dynamical System (eMLD), see Subsection 7.2.2, allows one to quickly simulate the manufacturing system under finite servers semantics. Only two steps of the eMLD are required to obtain the evolution of the system marking (see Figure 8.3). At the first step ($\tau = 8$) place p_4 becomes empty, at the second step ($\tau = 12$) place p_2 becomes empty too and the steady state is reached.

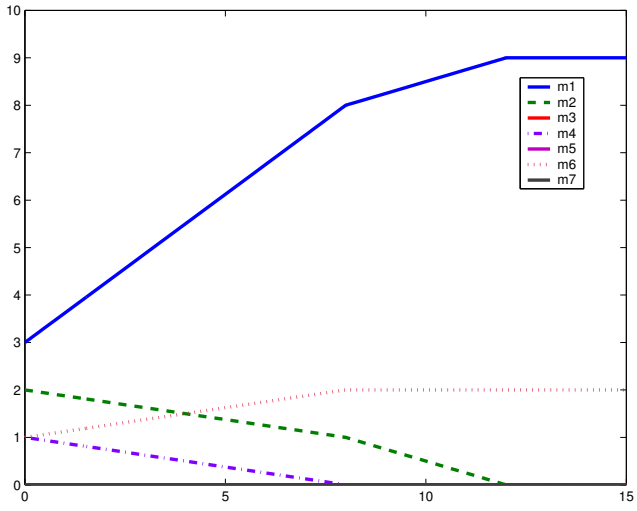


Figure 8.3: Evolution of the manufacturing system under finite servers semantics.

The ideas presented in Chapter 7 can be directly applied to the table factory to control its behaviour. Let us assume, for instance, that it is requested to maximize the throughput at the steady state. For that purpose a Mixed Integer Linear Programming (MILP) problem that forces the marking in the last two steps to be the same and maximizes the flow of a transition can be designed. Since the net is mono-T-semiflow, in the steady state the flow of the transitions will be proportional to the T-semiflow. Hence, in the objective function any transition can be taken to maximize the throughput. The solution of the MILP gives the following control for time $\tau = 0$: $\mathbf{u}[t_1](0) = 0$, $\mathbf{u}[t_2](0) = \mathbf{u}[t_3](0) = \mathbf{u}[t_4](0) = \mathbf{u}[t_5](0) = \mathbf{u}[t_6](0) = 1/3$. For that control law, the flow through transitions is $\mathbf{f}[t_1](0) = 1$, $\mathbf{f}[t_2](0) = \mathbf{f}[t_3](0) = \mathbf{f}[t_4](0) = \mathbf{f}[t_5](0) = \mathbf{f}[t_6](0) = 2/3$ that is proportional to the T-semiflow. In other

words, if the control is maintained the marking will remain constant indefinitely, i.e., in the steady state.

Let us now consider the time optimal control problem of reaching a marking in which $\mathbf{m}[p_3] = 0.4$ and $\mathbf{m}[p_4] = 0.75$ in minimum time, being transitions t_3 and t_4 the only controllable transitions. A boolean variable will be introduced in the MILP that will be true iff a target marking ($\mathbf{m}[p_3] = 0.4$ and $\mathbf{m}[p_4] = 0.75$) is reached. The MILP must explicitly specify that the boolean variable has to be true in the last step. The value to be minimized is the time elapsed till the boolean variable becomes true. The control law yielded by the MILP is $\mathbf{u}[t_1](0) = \mathbf{u}[t_2](0) = \mathbf{u}[t_4](0) = \mathbf{u}[t_5](0) = \mathbf{u}[t_6](0) = 0$, $\mathbf{u}[t_3](0) = 0.325$ during $q(0) = 2$ units of time. After that period of time the target marking is reached. Notice that with a greater $\mathbf{u}[t_3]$ a marking with $\mathbf{m}[p_3] = 0.4$ can be reached in less time than 2 time units. Nevertheless, the minimum time for place p_4 to reach 0.75 (once its output transition is completely opened, $\mathbf{u}[t_4](0) = 0$) is 2 time units.

8.2 An Assembly Line

The Petri net system in Figure 8.4 represents an assembly line with kanban strategy (see [ZRS01]). The system has two stages that are connected by transition t_{14}. The first stage is composed of three lines (starting from p_2, p_3 and p_4 respectively) and three machines (p_{23}, p_{24} and p_{25}). Places p_{26}, p_{27} and p_{28} are buffers at the end of the lines. The second stage has two lines that require the same machine/resource p_{18}. The number of kanban cards is given by the marking of places p_2, p_3 and p_4 for the first stage, and by the marking of p_{32} for the second stage. The system demand is given by the marking of p_1.

General considerations

The net has 12 minimal P-semiflows that cover every place, i.e., it is conservative. It also has one T-semiflow that is minimal and covers every transition, i.e., it is consistent. Thus, the net is mono-T-semiflow (Subsection 1.1.2). The net has eight transitions with two input places each ($|{}^\bullet t| = 2$), two transitions with three input places each ($|{}^\bullet t| = 3$) and one transition with four input places ($|{}^\bullet t| = 4$). This way, the number of structural PT-sets is $2^8 \cdot 3^2 \cdot 4 = 9 \cdot 2^{10}$. It is therefore remarkable that a continuous Petri net can represent such a large number of "embedded" linear systems in such a compact way. The other side of the coin is that a great effort is required if every linear system has to be studied separately, for example, to compute marking estimates (Section 6.3).

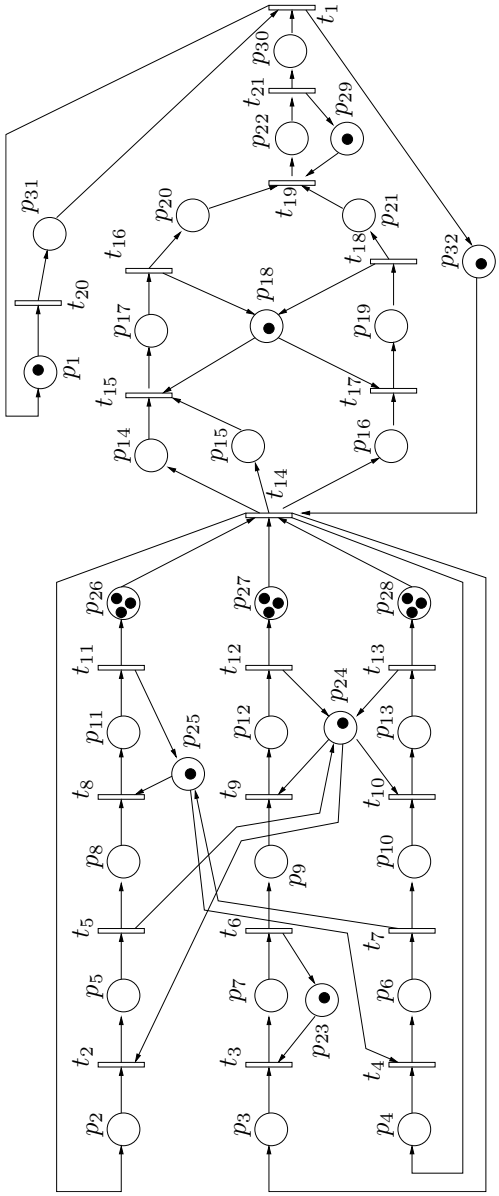


Figure 8.4: An assembly line with kanban strategy.

Reachability

Let the initial marking of the system be $\mathbf{m_0}[p_1] = \mathbf{m_0}[p_{18}] = \mathbf{m_0}[p_{23}] = \mathbf{m_0}[p_{24}] = \mathbf{m_0}[p_{25}] = \mathbf{m_0}[p_{29}] = \mathbf{m_0}[p_{32}] = 1$, $\mathbf{m_0}[p_{26}] = \mathbf{m_0}[p_{27}] = \mathbf{m_0}[p_{28}] = 3$ and the

marking of the rest of places be equal to zero. Every trap of the net is initially marked. Moreover, every trap contains a minimal P-semiflow, hence, emptying a trap implies emptying a P-semiflow. Since the net is consistent and conservative by Proposition 5.1 no solution of the state equation can yield a marking in which a P-semiflow (and therefore a trap) is empty. This way, the last condition in Theorem 2.12 for a marking to be reached can be dropped. Furthermore, the second condition can also be overlooked because the system is consistent and every transition is fireable (reasoning in a similar way to Corollary 2.16). Hence, the set of reachable markings with a finite firing sequence $\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ is equal to the set of solutions of the state equation, $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \mathrm{LRS}(\mathcal{N}, \mathbf{m_0})$. By Corollary 2.18 it holds $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \text{lim-}\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \delta\text{-}\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \mathrm{LRS}(\mathcal{N}, \mathbf{m_0})$, i.e., there are no spurious solutions of the state equation under any reachability concept.

Liveness

The net is mono-T-semiflow, thus, deadlock-freeness and liveness are equivalent. Let us assume that the system can reach a deadlock marking $\mathbf{m}$, i.e., the flow of every transition at $\mathbf{m}$ is zero. The PT-set associated to $\mathbf{m}$ must fulfill that the marking of all its places is zero. Nevertheless, it can be checked that each of the $9 \cdot 2^{10}$ structural PT-sets contains at least one P-semiflow. Since all P-semiflows are initially marked, there is no marking solution of the state equation at which there is an empty P-semiflow (Proposition 5.1). Therefore, the system is live both as untimed (under any reachability concept) and as timed (under any time interpretation). This implies that the system is also structurally live as discrete [Rec98]. The condition of Theorem 4.13 for robust-liveness necessarily holds.

Performance Evaluation

The only T-semiflow is a vector of ones. This implies that in the steady state the throughput of all the transitions will be the same. Let us assume that the firing rates of the transitions are $\boldsymbol{\lambda}[t_2] = \boldsymbol{\lambda}[t_3] = \boldsymbol{\lambda}[t_4] = \boldsymbol{\lambda}[t_8] = \boldsymbol{\lambda}[t_9] = \boldsymbol{\lambda}[t_{10}] = \boldsymbol{\lambda}[t_{14}] = \boldsymbol{\lambda}[t_{15}] = \boldsymbol{\lambda}[t_{17}] = \boldsymbol{\lambda}[t_{19}] = \boldsymbol{\lambda}[t_{20}] = 10$, $\boldsymbol{\lambda}[t_1] = \boldsymbol{\lambda}[t_5] = \boldsymbol{\lambda}[t_6] = \boldsymbol{\lambda}[t_7] = \boldsymbol{\lambda}[t_{11}] = \boldsymbol{\lambda}[t_{12}] = \boldsymbol{\lambda}[t_{13}] = \boldsymbol{\lambda}[t_{16}] = \boldsymbol{\lambda}[t_{18}] = \boldsymbol{\lambda}[t_{21}] = 1$. Let us first consider the system as discrete with markovian infinite servers timing at transitions. Let the initial marking be $\mathbf{m_0}[p_1] = \mathbf{m_0}[p_{18}] = \mathbf{m_0}[p_{23}] = \mathbf{m_0}[p_{24}] = \mathbf{m_0}[p_{25}] = \mathbf{m_0}[p_{29}] = \mathbf{m_0}[p_{32}] = 1$, $\mathbf{m_0}[p_{26}] = \mathbf{m_0}[p_{27}] = \mathbf{m_0}[p_{28}] = 3$ and the marking of the rest of places be equal to zero. The size of the reachability set is 209704. The solution of the associated Markov chains yields a throughput of 0.2241 for every transition. The throughput bound for the fluidified system under infinite servers semantics (computed according to LPP (5.19)) is 0.3030, being $\{p_{14}, p_{17}, p_{20}, p_{22}, p_{30}, p_{32}\}$ the bottleneck P-semiflow. It can be checked that every T-coverture of the system contains a P-semiflow. Thus,

by Corollary 5.6, the throughput (0.3030) obtained with LPP (5.19) is the exact throughput of the continuous system in the steady state. Table 8.2 compares the throughput of the discrete system with the throughput of the fluidified one for two initial markings of p_{32}.

$\mathbf{m_0}[p_{32}]$	Reachable markings	$\chi[t_1]$ discrete	$\mathbf{f}_{ss}[t_1]$ continuous	Relative error
1	209704	0.2241	0.3030	26.04%
2	953200	0.2888	0.3226	10.73%

Table 8.2: Throughput of the discrete system, throughput of the continuous system and relative error.

The size of the reachability set of the discrete system is very sensitive to changes in the initial marking of $\mathbf{m}[p_{32}]$. In fact, if the initial marking of p_{32} is 2 instead of 1 the size of the reachability set is 953200, and if $\mathbf{m_0}[p_{32}] = 3$ the the number of reachable markings becomes 2859600. Such large reachability sets pose serious computational problems if one desires to compute the throughput of the discrete system. However, the throughput in the steady state of the fluidified system can be obtained in polynomial time thanks to LPP (5.19), since the condition in Corollary 5.6 verifies. The throughput of the continuous system can be considered as an "estimate" for the throughput of the discrete system (it is not necessarily un upper bound for the throughput of the discrete system, see Subsection 5.1.1). The throughput of the continuous system in the steady state with $\mathbf{m_0}[p_{32}] = 2$ is 0.3226 (in comparison with a throughput of 0.28883 if the Markov chain of the discrete system is solved). The bottleneck P-semiflow for the continuous system is $\{p_5, p_{12}, p_{13}, p_{24}\}$. The same throughput (with the same bottleneck P-semiflow) of the continuous system is obtained if $\mathbf{m_0}[p_{32}] = 3$.

Observability

Let us assume that the system is required to be fully observable whatever the PT-set ruling its evolution is. To achieve this requirement every potential timed-implicit place (Subsection 1.3.2) must be measured; if a place is timed-implicit the system dynamics does not depend on it and its marking cannot be computed unless it is directly measured (Section 6.1). Every place p in a synchronization (p has an output transition t such that $|{}^\bullet t| > 1$) can be timed-implicit if its marking does not constrain its output transitions. Thus, in order to have a fully observable system at least every place in a synchronization must be measured.

In the assembly line system, the places that are in a synchronization are p_2, p_3, p_4, p_8, p_9, p_{10}, p_{14}, p_{15}, p_{16}, p_{18}, p_{20}, p_{21}, p_{23}, p_{24}, p_{25}, p_{26}, p_{27}, p_{28}, p_{30}, p_{31} and

p_{32}. It turns out that if all these places are measured then the system becomes fully observable not only for any initial marking ($\mathbf{m_0}$) but also for any vector of firing rates ($\boldsymbol{\lambda}$). This is similar to the structural observability for Join Free systems described in Subsection 6.2.1. This full observability is easy to check if one focuses on the non measured places and tries to compute their markings separately (following the same reasoning as in Subsection 6.2.1). Let us, for example, focus on the non measured place p_{22}. Since places p_{30} and p_{31} are measured, the flow of transition t_1 is easy to obtain. Notice that the derivative of the marking of p_{30} is equal to flow of t_{21} minus the flow of t_1. Given that the flow of t_1 can be obtained and the marking of p_{30} is measured, one can compute the flow of t_{21}. Since the only input place of t_{21} is p_{22}, the flow of t_{21} is proportional to the marking of p_{22} ($\mathbf{f}[t_{21}] = \boldsymbol{\lambda}[t_{21}] \cdot \mathbf{m}[p_{22}]$) and $\mathbf{m}[p_{22}$ can be immediately computed. The same reasoning can be applied to the rest of non measured places.

8.3 A Car Traffic System

The main concern of this section is to model a car traffic system with continuous Petri nets. The model should be able of representing the various modes of operation of a traffic system: Free flow traffic, saturated/congested traffic, traffic jams, stop-and-go waves, etc. The use of traffic models gives one the chance of analyzing, predicting and controlling the behaviour of traffic systems. This section introduces a new time interpretation that is useful to obtain "realistic" and yet compact models for traffic systems [JB05].

Traffic systems are discrete systems that are typically heavily populated. Instead of discrete models, continuous models can be used to macroscopically model the behaviour of traffic systems. In the scope of traffic systems, macroscopic models (see [HB01, KPD$^+$02, Hel97]) disregard individual cars and consider mainly three real valued variables: density, speed and flow. The road network to be modelled is subdivided into sections. Each road section is modelled separately by a continuous Petri net. The model for the whole network is obtained by joining the nets for the sections: The flow of cars from one section to the next section is represented by the flow of the transition interconnecting both sections. This way, the model becomes highly compositional. Combining these continuous PN models with discrete PN models of traffic lights leads to a hybrid Petri net model. In order to distinguish between continuous and discrete net elements, a discrete place/transition will be represented as a circle/line, and a continuous place/transition will be represented as a double circle/white box.

The following subsections establish some necessary concepts to model a a traffic system with continuous Petri nets. In the last subsection, these concepts are used to model and simulate the behaviour of an intersection regulated by traffic lights.

Some modelling considerations

Infinite servers semantics is used in system models in which the processing speed, i.e., flow of transitions, is proportional to the number of customers in the upstream place, i.e., enabling degree. The following examples show how the flow of transitions and the rate of change of the marking of places can be affected by the arc weights.

Consider transition t_1 (see Figure 8.5(a)) that has one input place p_1. Its flow is $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] \cdot \mathbf{m}[p_1]/z$ where $z > 0$ is the weight of the arc. As shown above under infinite servers semantics the marking changes according to $\dot{\mathbf{m}}[p_1] = -z \cdot \mathbf{f}[t_1] = -\boldsymbol{\lambda}[t_1] \cdot \mathbf{m}[p_1]$. Thus, the evolution of the marking of p_1 does not depend on z, i.e., the weight of the arc.

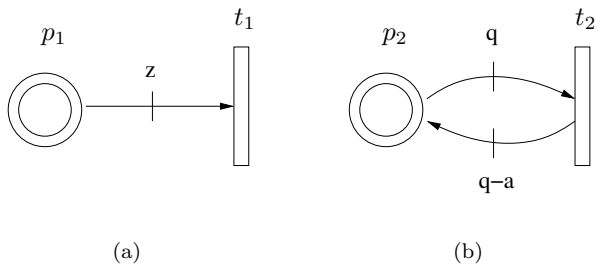


Figure 8.5: Two modelling options.

By slightly manipulating the system in Figure 8.5(a) it is possible to obtain a system in which the evolution of p_1 depends on the weight of its input (output) arc. Consider the system in Figure 8.5(b) with $q > 0$ and $q - a > 0$ since arc weights must be positive. The flow of the transition is $\mathbf{f}[t_2] = \boldsymbol{\lambda}[t_2] \cdot \mathbf{m}[p_2]/q$ and the marking of p_2 evolves according to $\dot{\mathbf{m}}[p_2] = (q - a - q) \cdot \mathbf{f}[t_2] = -a/q\boldsymbol{\lambda}[t_2] \cdot \mathbf{m}[p_2]$, depending on the parameter values q and a. If $a > 0$ the marking of p_2 decreases (due to the constraint $q - a > 0$ the maximum rate of decrease is bounded by $\dot{\mathbf{m}}[p_2] = -\boldsymbol{\lambda}[t_2] \cdot \mathbf{m}[p_2]$). If $a = 0$ then $\mathbf{m}[p_2]$ is constant and so is the flow of t_2. If $a < 0$ then $\mathbf{m}[p_2]$ increases.

Following these ideas the flow of a transition can be modelled as a piecewise linear function of the marking of a given place. For example, the system in Figure 8.6(a) uses four loops of arcs to model the flow of t_1 with respect to the marking of p_1. The thick line in Figure 8.6(b) shows the piecewise linear dependance of $\mathbf{f}[t_1]$ on $\mathbf{m}[p_1]$ (λ is the firing rate of t_1). The existence of P-semiflows greatly helps to obtain this modelling capability.

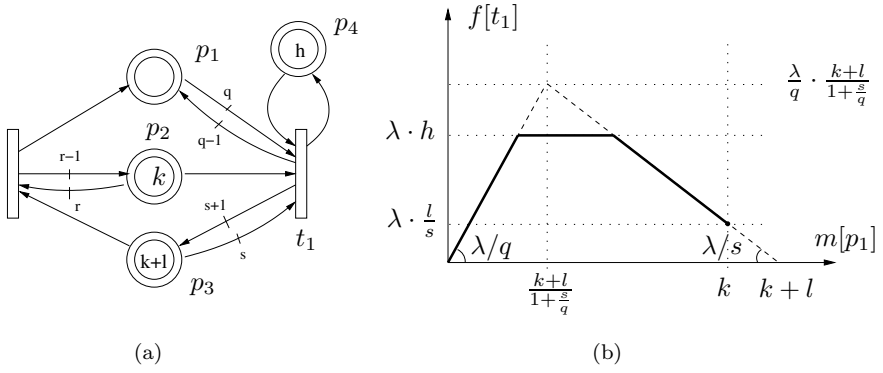


Figure 8.6: The flow of transition t_1 is a piecewise linear function of the marking of p_1.

Discrete time model

The system in Figure 8.7 represents a machine, t_1, working at constant speed, $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] \cdot \mathbf{m}[p_1]$, that places its production in a conveyor, p_2. One can imagine that machine t_1 places pieces of finished material at uniformly distributed locations on the conveyor belt p_2, that moves those pieces to the second machine t_2. Machine t_2 processes its input material and stores it in the warehouse p_3. The initial marking of the system is $\mathbf{m_0} = (1\ 0\ 0)$, i.e., the conveyor and the warehouse are initially empty.

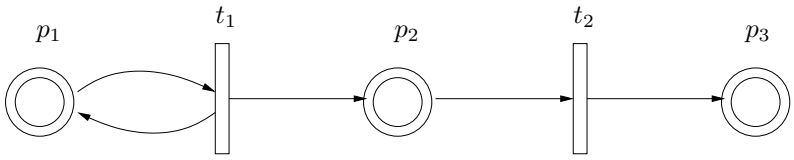


Figure 8.7: A continuous Petri net modelling a conveyor.

According to the usual continuous time model the initial flow of t_1 is $\mathbf{f}[t_1](\tau = 0) = \boldsymbol{\lambda}[t_1]$. This is implies that material is placed in the conveyor p_2 from the initial instant $\tau = 0$ ($\mathbf{m}[p_2](\tau) > 0$ for every $\tau > 0$). This entails $\mathbf{f}[t_2](\tau) > 0$ for every $\tau > 0$. This behaviour cannot be a faithful representation of the real system behaviour since it implies that the material has spent zero units of time to reach t_2, i.e., the conveyor is infinitely fast (or infinitely short).

One way of avoiding an infinitely fast movement of material going from one transition to the next one is using a discrete time model. According to this model, time is discretized in steps (intervals) of length $\Delta > 0$. At the beginning of each step the flow of the transitions is computed with the usual expression for infinite servers

semantics: $\mathbf{f}[t](k) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t}\{\mathbf{m}[p](k)/\mathbf{Pre}[p,t]\}$ for the k^{th} step. The marking at the next step is defined by $\mathbf{m}(k+1) = \mathbf{m}(k) + \mathbf{C} \cdot \mathbf{f}(k) \cdot \Delta$. This way, the flow of a transition during Δ units of time depends only on the marking of its input places at the beginning of the interval. The interval Δ can be seen as the travelling time (delay) of the material between two transitions. In Figure 8.7, Δ is the time the conveyor takes to arrive from t_1 to t_2. Notice that the flow of t_2 is zero during the first interval ($\mathbf{f}[t_2](\tau = 0..\Delta) = 0$ if the system is discrete time, $\mathbf{f}[t_2](\tau) > 0$ for every $\tau > 0$ if it is continuous time).

This discrete time continuous PN model makes it possible to represent delays; moreover the fact that the flow of the system is constant during each interval allows fast simulations.

In the discrete time model Δ is a design parameter. The larger Δ is the longer the delays one can model easily. However taking Δ too large can lead to negative markings since the marking may be linearly decreasing during an interval. Fortunately, it is possible to compute an upper bound for Δ in order to ensure the nonnegativeness of the marking. Such upper bound depends only on the structure of the net (not on the marking). In order to compute this upper bound, each place will be considered separately. It will be assumed that no input flow is coming into the place and it will be computed how fast it can become empty. The interval required to empty the place that takes the shortest time to become empty is the upper bound.

Let us compute how fast the place p_1 of the system in Figure 8.8(a) can become empty. Clearly, the marking of p_1 decreases iff $r > s$, hence only this case is considered. Let us first compute how long it takes to empty p_1 if $\mathbf{m}[p_1]/r \leq \mathbf{m}[p_2]/q$, i.e., $\mathbf{m}[p_1]$ defines the enabling degree of t_1. In that case $\mathbf{f}[t_1](k) = \boldsymbol{\lambda}[t_1] \cdot \mathbf{m}[p_1](k)/r$ and $\mathbf{m}[p_1](k+1) = \mathbf{m}[p_1](k) + (s-r) \cdot \boldsymbol{\lambda}[t_1] \cdot \mathbf{m}[p_1](k)/r \cdot \delta$. It follows that $\mathbf{m}[p_1](k+1) = 0$ when $\delta = r/(\boldsymbol{\lambda}[t_1] \cdot (r-s))$. Notice that in the case that $\mathbf{m}[p_1]/r > \mathbf{m}[p_2]/q$ ($\mathbf{m}[p_2]$ defines the enabling degree) the flow through t_1 would be less than in the previous case and therefore it would take longer to empty p_1. For the system in Figure 8.8(a), the shortest time to empty p_1 is $r/(\boldsymbol{\lambda}[t_1] \cdot (r-s))$. Any Δ smaller than $r/(\boldsymbol{\lambda}[t_1] \cdot (r-s))$ prevents p_1 from becoming negative.

A similar approach can be taken to compute a bound for Δ for a system having places with several output transitions (see Figure 8.8(b)). As in the previous example, in order to compute the fastest emptying time of p_1 only the output transitions that decrease the marking are considered, i.e., t_1 (t_2) is considered iff $r > s$ ($u > v$). The shortest emptying time occurs when p_1 is determining the flow of both output transitions. For a general system with several places, in order to avoid negative markings the value of Δ has to be at most:

$$\min_{p, \exists t \in\, p^\bullet, \mathbf{Pre}[p,t] > \mathbf{Post}[p,t]} \left\{ \frac{1}{\sum_{t \in\, p^\bullet,\ \mathbf{Pre}[p,t] > \mathbf{Post}[p,t]} \dfrac{\boldsymbol{\lambda}[t] \cdot (\mathbf{Pre}[p,t] - \mathbf{Post}[p,t])}{\mathbf{Pre}[p,t]}} \right\}$$

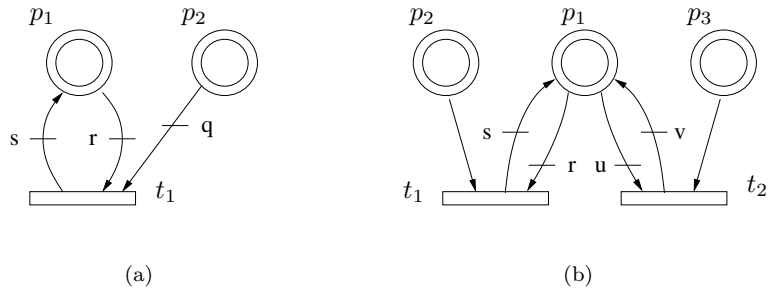


Figure 8.8: The value of Δ is upper bounded.

Notice that this expression depends only on the structure of the net and not on the initial marking.

Emptying places in finite time

Let us consider the discrete time evolution of the system in Figure 8.9. Let Δ be the length of the time interval of the discrete time model (according to the previous Subsection, Δ is upper bounded by $\min\{1/\boldsymbol{\lambda}[t_1], 1/\boldsymbol{\lambda}[t_2]\}$). After the first interval, the marking of p_1 is $\mathbf{m}[p_1](1) = \mathbf{m}[p_1](0) + \mathbf{C}\cdot\mathbf{f}[t_1](0)\cdot\Delta = \mathbf{m}[p_1](0) - \boldsymbol{\lambda}[t_1]\cdot\mathbf{m}[p_1](0)\cdot\Delta = (1-\boldsymbol{\lambda}[t_1]\cdot\Delta)\cdot\mathbf{m}[p_1](0)$. After the second interval $\mathbf{m}[p_1](2) = (1-\boldsymbol{\lambda}[t_1]\cdot\Delta)\cdot\mathbf{m}[p_1](1) = (1-\boldsymbol{\lambda}[t_1]\cdot\Delta)^2\cdot\mathbf{m}[p_1](0)$ and after the k^{th} interval $\mathbf{m}[p_1](k) = (1-\boldsymbol{\lambda}[t_1]\cdot\Delta)^k\cdot\mathbf{m}[p_1](0)$. This way, if $\Delta = 1/\boldsymbol{\lambda}[t_1]$, p_1 becomes empty after the first step and remains empty indefinitely. However, if $\Delta < 1/\boldsymbol{\lambda}[t_1]$ the evolution of $\mathbf{m}[p_1]$ follows a geometric progression and never gets completely empty.

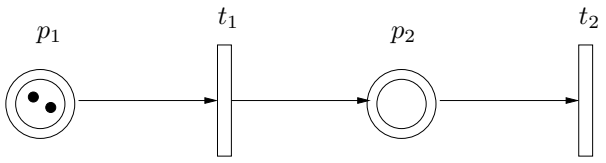


Figure 8.9: p_1 is emptied in finite time iff $\Delta = 1/\boldsymbol{\lambda}[t_1]$.

From a modelling point of view a geometrical emptying of a place can be useful, for example, to model how a capacitor discharges exponentially. Nevertheless, for other modelling purposes this feature is not desired. Suppose that the marking of p_1 is the number of customers waiting to be served by t_1, and t_1 is a server that starts working at a speed that is proportional to the length of the queue. If there are no new customers coming into the queue the speed of t_1 should remain constant until

the queue empties. It makes no sense that the speed of t_1 decreases as the queue gets shorter.

By slightly modifying the described firing semantics it is possible to avoid falling in a geometric progression when emptying a place: For a given transition t and at a given step k it will be checked whether its input place determining the enabling degree had input flow (new customers) during the previous step $k-1$. If there was no input flow to that place the flow of t is kept the same, $\mathbf{f}[t](k) = \mathbf{f}[t](k-1)$, otherwise the usual firing semantics is applied, $\mathbf{f}[t](k) = \boldsymbol{\lambda}[t] \cdot \text{enab}(t, \mathbf{m}, k)$. Keeping the same flow of a transition allows one to empty a place in finite time. Anyway, one should be aware that this modification in the model leads to *non pure* discrete time infinite servers semantics and could cause negative markings even if the bound for Δ is considered. In order to avoid negative markings, the flow of the transitions will be forced to be the minimum between the value just described and the flow that would empty one of the input places at the end of the time interval. This way, places become empty exactly at the end of time intervals.

The reader should observe that many properties that make Petri nets so useful for modelling remain valid after the modifications introduced in this section. For example our discrete time PNs will satisfy place and transition invariants, markings are still states for the dynamic evolution, structural analysis is applicable, etc.

Modelling a road section

The traffic model to be presented requires a spatial discretization of the road to be modelled, i.e., the road is divided into several sections. In this subsection, a continuous PN model of one single road section is presented. Subsequently, this model will be used as a building block for representing large networks.

The state of a section of a road network is described by three macroscopic variables: Density of cars, average speed and flow. The marking m of a place will represent the number of cars in the section, these cars being uniformly distributed along the length of the section, and having average speed v. Note that m is proportional to the density d of cars along the section. The flow f of cars leaving the section is then $f = d \cdot v$.

In a traffic system the cars in a section with low density travel at a given free speed, *free flow traffic*. Hence, the flow out of the section increases proportionally to the density. When the density of the section is higher, the average speed decreases and the flow out of the section keeps *ideally constant*. If the density is much higher, the traffic becomes heavy and the flow out of the section decreases. This (bell shape) relationship between the flow and the density is known as the fundamental traffic diagram. First, a net that models free flow traffic and constant flow traffic is presented. Later on, it will be shown how the decreasing of the flow is modelled when the density is high.

The number of cars in road section i will be represented by the marking of a place, p_1^i in Figure 8.10(a), and the flow of cars leaving the section will be the flow of a transition, t_i. If p_3^i is ignored, the use of infinite servers semantics establishes $\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \mathbf{m}[p_1^i]$, i.e., the outflow is proportional to the density. Hence, the subnet p_1^i, t_i with an appropriate $\boldsymbol{\lambda}[t_i]$ models free flow traffic. Notice that this relationship between the flow and the marking, $\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \mathbf{m}[p_1^i]$, cannot be modelled with finite servers semantics in which the flow of a transition is independent of the marking of its positively marked input places.

Constant flow traffic can be modelled by adding p_3^i. The marking of p_3^i is always constant and imposes an upper bound on the flow of t_i, $\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \min\{\mathbf{m}[p_1^i], \mathbf{m}[p_3^i]\}$. Therefore, when $\mathbf{m}[p_1^i] > \mathbf{m}[p_3^i]$ the flow of t_i is constant, $\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \mathbf{m}[p_3^i] = \boldsymbol{\lambda}[t_i] \cdot h^i$.

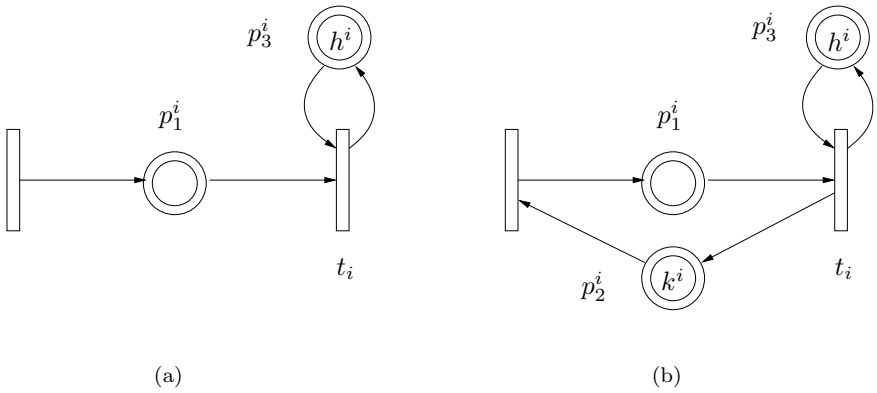


Figure 8.10: Modelling a section.

Obviously, the number of cars that can be in a road section is finite. This means that the model of a section must impose an upper bound on the marking of the place representing the number of cars. This can be easily achieved by adding a new place to the section model, see p_2^i in Figure 8.10(b). At any time it holds $\mathbf{m}[p_1^i] + \mathbf{m}[p_2^i] = k^i$ where k^i represents the capacity of the section and $\mathbf{m}[p_2^i]$ the number of free gaps.

Joining sections

In a Petri net model with several sections, two adjacent sections, i, j, share a transition, t_i, whose flow represents the number of cars passing from section i to section j per time unit. Hence, a given transition t_i of the net model has three input places: p_1^i representing the number of cars in section i, p_3^i with constant marking bounding the flow of t_i and p_2^j representing the number of gaps in section j. Therefore, the flow of

cars from section i to section j also depends on the number of gaps in the downstream section j, $\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \min\{\mathbf{m}[p_1^i], \mathbf{m}[p_3^i], \mathbf{m}[p_2^j]\}$. This fact is very realistic, if one considers for example, how a traffic jam (decreasing of the flow when the density is high) propagates from downstream to upstream sections. The flow of t_i is the minimum between the number of cars desiring to leave the section (sending function) and the number of cars allowed to enter the next section (receiving function) [Dag95].

The outflow from a low dense section i (free flow) is proportional to the number of cars ($\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \min\{\mathbf{m}[p_1^i]\}$) being $\boldsymbol{\lambda}[t_i]$ the proportionality constant. If the downstream section becomes full, the outflow is proportional to the number of gaps of the downstream section ($\mathbf{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \min\{\mathbf{m}[p_2^j]\}$) with $\boldsymbol{\lambda}[t_i]$ as the proportionality constant. That is, the proportionality constant, $\boldsymbol{\lambda}[t_i]$, is the same under both situations. One way to avoid this fact is to use the arc loops presented in Figure 8.5. The use of such arc loops allows one to have different proportionality constants for the density of cars and the number of gaps. Notice that the constant flow traffic is modelled thanks to a place, p_3^i, with a constant marking that does not represent any real amount. Hence, for any $\boldsymbol{\lambda}[t_i]$ its marking can be chosen to correctly upper bound the flow of t_i without introducing weights in its input/output arcs. Figure 8.11 shows a traffic model consisting of three sections and arc loops to control the proportionality constants. With an appropriate $\boldsymbol{\lambda}$, that system can be reduced to an equivalent one with only one arc loop on each transition ($\boldsymbol{\lambda}[t_i]$ is already the proportionality constant either for the density or for the number of gaps).

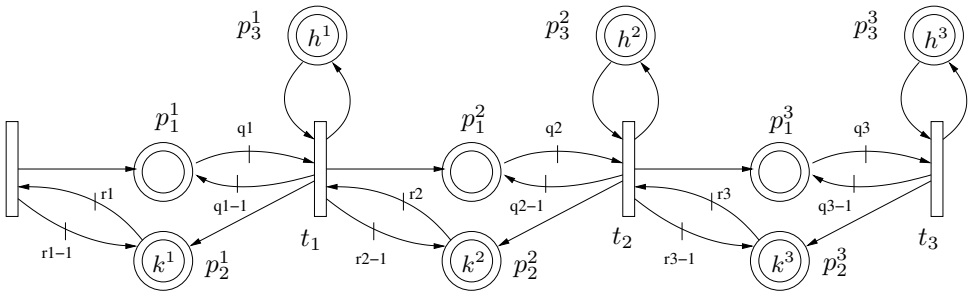


Figure 8.11: Traffic system with three sections.

Notice that the special features of the model described previously are useful for traffic modelling. By using a discrete time model it is possible to represent the delay of the cars coming from the input transition of one section to the output transition. The time interval, Δ, can be seen as the time required for a car to travel from the beginning of a section to the beginning of the next one. A continuous time model could be possible, but then it would require an infinite dimensional state space, corresponding to infinitely short sections (a partial differential equation is obtained by letting Δ go

to 0). Besides, the extensions presented allow sections to become empty in finite time by keeping the outflow constant as long as no inflow exists. This extensions represent more faithfully the behaviour of a real traffic system than the original continuous Petri net formalism and will be used in the sequel.

Simulating traffic behaviour in an intersection

The most usual way to control real traffic systems is through traffic lights. Traffic lights can be seen as a discrete event system whose state can be either red, amber or green. In our model traffic lights are modelled as a discrete Petri net, see Figure 8.12 for traffic lights ruling an intersection with two crossing lanes $L1$ and $L2$.

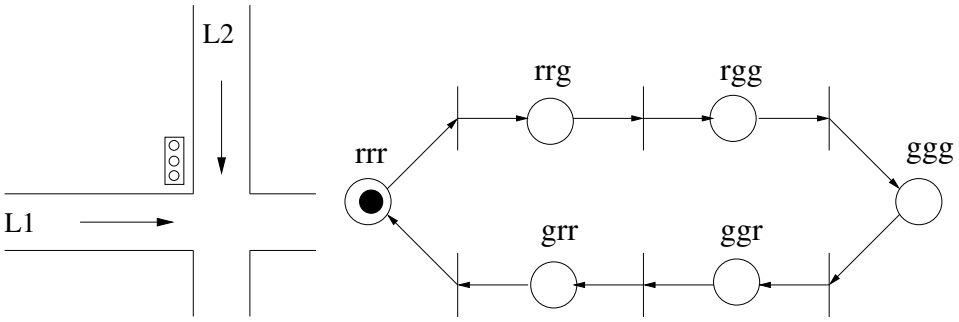


Figure 8.12: A discrete Petri net modelling traffic lights in an intersection.

The system that models traffic lights has six phases represented by each of the places of the net. A given phase is active when its corresponding place is marked. Since only one phase can be active at a given instant, the number of tokes in the net is 1. The meaning of the phases is: *ggg*: cars of $L1$ crossing, *ggr*: stopping traffic of $L1$, *grr*: cars of $L2$ start crossing, *rrr*: cars of $L2$ crossing, *rrg*: stopping traffic of $L2$, *rgg*: cars of $L1$ start crossing. The use of the phases *ggr*, *grr*, *rrg*, *rgg* allows one a more realistic modelling of the system since they model how the flow of cars softly becomes either zero or positive.

Figure 8.13 models four sections and an intersection in which the traffic is regulated by a traffic lights system like the one in Figure 8.12. It is a hybrid Petri net since it includes discrete and continuous places and transitions. When the traffic lights system is at *ggg* the flow of t_1 depends on the marking of its input places and the flow of t_2 is 0. Similarly, at phase *rrr* t_1 is blocked and the flow of t_2 depends on the marking of its input places. For the rest of the phases, *ggr*, *grr*, *rrg*, *rgg*, the flow

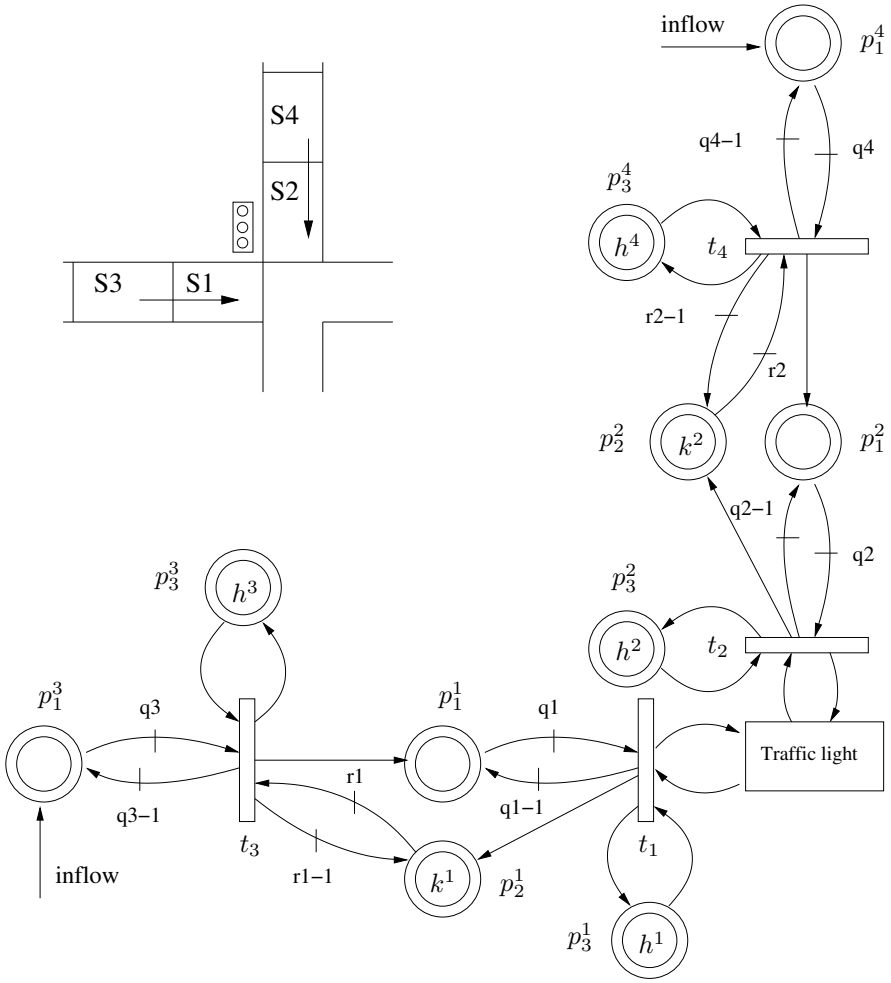


Figure 8.13: A Petri net modelling an intersection.

of the transitions involved in the intersection, t_1 and t_2, is defined to model how the flow of cars speeds up and slows down when the traffic lights system switches: If it switches to green (*rgg* for section 1) the flow of t_1 increases linearly from zero to the flow computed with the usual infinite firing semantics; if it switches to red (*ggr* for section 1) the flow of t_1 decreases linearly to zero. These behaviours can be easily simulated by computing a constant flow that produces the marking that would be obtained if the flow increased/decreased linearly.

Consider the system in Figure 8.13. Let the capacity of the sections be 80 cars ($\mathbf{m}[p_1^i] + \mathbf{m}[p_2^i] = 80$ for $i = 1 \ldots 4$), $\boldsymbol{\lambda}[t_i] = 4$ for $i = 1 \ldots 4$, $q^i = 100$ for $i = 1 \ldots 4$,

$r^i = 80$ for $i = 1, 2$, $h^i = 0.5$ for $i = 1 \dots 4$, the initial load of the sections $\mathbf{m}[p_1^1] = 50$, $\mathbf{m}[p_1^2] = 30$, $\mathbf{m}[p_1^3] = 35$, $\mathbf{m}[p_1^4] = 60$ and the time step $\Delta = 8$ seconds. Let us assume that the traffic lights system is initially red for section 1 and that there exist constant input flows of 0.8 cars per second entering section 3 and 0.5 cars per second entering section 4. Once all these initial conditions are established, the traffic model can be easily simulated to predict the traffic behaviour in the intersection.

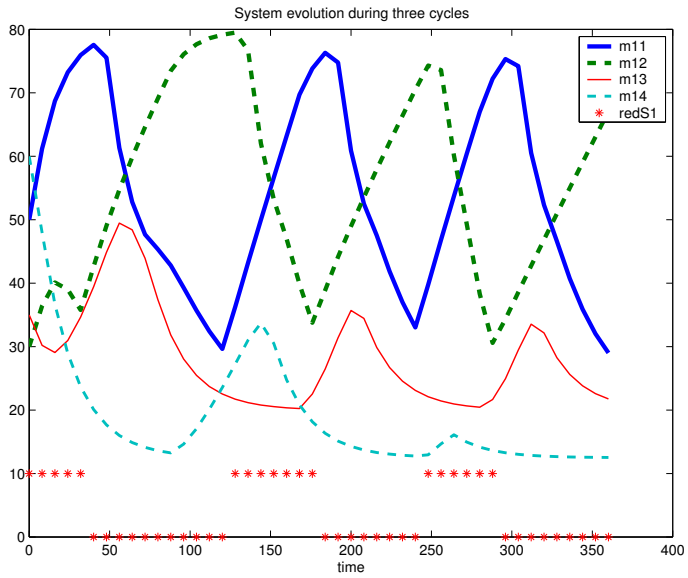


Figure 8.14: Evolution of the system in Figure 8.13.

Figure 8.14 depicts one possible evolution the traffic system. The plot covers three traffic lights cycles of length 120 seconds each. In the figure, stars at a level of 10 mean red light for section 1. Obviously, the evolution of the system strongly depends on the switching times of the traffic lights.

Traffic models can be useful for different purposes. As shown in this section they can provide a prediction of a given traffic system scenario. Predicting the behaviour of a traffic system through a model can be useful to ease the decision making process that intends to alleviate potential traffic jams: Traffic lights should be controlled in such a way that the total delay of the vehicles in the system is minimized. Other possible control strategies are to minimize the total fuel consumption, maximize the throughput of a given road section, etc. In this framework, continuous Petri nets turn out to be a very suitable choice to cope with the task of modelling traffic systems: The model can be built in a highly compositional and intuitive way and simulations can be carried out very fast.

Concluding Remarks

Fluidification is a relaxation technique aiming to avoid the state explosion problem of large discrete systems. Roughly, fluidifying a discrete system means that the discrete variables of the system are converted into real variables. Such a relaxation offers the possibility of using linear instead of integer techniques to check some system properties. This usually implies that the complexity of verification tasks is reduced, even to *polynomial* time. Unfortunately, the properties of the original discrete system are not always preserved by its fluidified version, i.e., there can exist a "significant" gap between the behaviour of the discrete system and the behaviour of the fluidified one. This fact motivates the study of fluidified systems in order to establish which their properties are and which discrete systems are suitable to be fluidified.

The fluidification of Petri nets results in continuous Petri nets. In a continuous Petri net the amount in which a transition is fired is not restricted to the natural numbers but to the positive real numbers. Such transition firings produce a continuous net marking that is represented by a vector of non negative real numbers.

This work has dealt with both untimed and timed continuous Petri nets. In untimed nets there exists no a priori policy for the firing of the transitions, i.e., there exists a nondeterminism in the evolution of the system. On the contrary, a time interpretation is adopted for timed net systems. The time interpretation of a net system completely determines its marking evolution along time. Two different time interpretations have been considered for the firing of transitions in continuous nets: Infinite servers semantics and finite servers semantics. Under both semantics the marking evolution can be seen as the evolution of a piecewise linear system. Moreover, a third time interpretation has been introduced in Section 8.3 in order to model in a "realistic" way a car traffic system.

Some *unexpected* results regarding the fluidification of untimed and timed Petri nets have been shown along this work: Liveness of the discrete system is not a necessary nor a sufficient condition for liveness of the continuous system, this already known result for untimed systems [RTS99] also applies to timed systems. Although fluidifying means "removing firing constraints", the performance of the continuous system is not necessarily better than the performance of the discrete one.

Since in continuous Petri nets the transitions are fired in real amounts, the set of reachable markings is a region in the continuous space. However, the concept of reachability is not as straightforward as in discrete nets. Chapter 2 deals with three different concepts of reachability for untimed continuous nets: *reachability*, lim-*reachability*, δ-*reachability*, the third one being a contribution of this work. These concepts differ in the length of the firing sequence, finite vs. infinite, and in the way a marking is reached, a marking is effectively reached vs. the system gets as close to the marking as desired. *Reachability* stands for those markings that are effectively reached with a finite firing sequence. The concept of lim-*reachability* applies to those markings that are effectively reached with a finite or infinite firing sequence. Clearly, the set of *reachable* markings is contained in the set of lim-*reachable* markings. A marking is said to be δ-reachable if the system can get as close as desired to it with a finite firing sequence. The set of lim-reachable markings is contained in the set of δ-reachable markings. It could be thought that a fourth reachability concept should exist: According to that fourth concept a marking would be reachable if the system could get as close as desired to it with a finite or infinite firing sequence. This fourth concept is, however, equivalent to δ-*reachability*: Converging with an infinite sequence to a marking m' that is as close as desired to a target marking m is equivalent to reaching with a finite sequence a marking m'' that is as close as desired to marking m.

The reachability sets according to all three reachability concepts have been fully characterized. The fundamental state equation turns out to be an essential instrument for that characterization. The more relaxed the reachability concept is the easier its characterization is obtained. This way, the set of *reachable* markings is the hardest to be characterized and the set of δ-*reachable* the easiest. To characterize the set of *reachable* markings, traps, fireability of the firing sequence and the state equation have to be considered. For lim-*reachability*, traps do not have to be considered. For δ-*reachability* only the state equation is necessary. Thus, there exist no spurious solutions of the state equation under δ-*reachability*. Furthermore, the differences (if any) among the three reachability sets lay only in the border points of the set of δ-*reachable* markings.

Liveness has been studied for untimed and timed mono-T-semiflow Petri nets. Since a mono-T-semiflow net has only one T-semiflow every potentially live infinite behaviour is constrained to that T-semiflow, i.e., in an infinite horizon the transitions have to be fired in the proportions specified by the T-semiflow. Hence, if the structure of the net does not force the fact that the transitions are fired according to the T-semiflow the system will deadlock. With respect to untimed net systems (Chapter 3), if the set of input places of a transition t is contained in the set of another transition t' then the system is not live. This is an easy to check structural condition that does not depend on the weight of the arcs, it is actually a topological condition. This

derives from the fact that in continuous net systems the weight of an arc determines the maximum amount in which a transition can be fired but never prevents it from firing.

In timed net systems liveness refers to the behaviour of the system at the steady state: A system is live iff the vector of flows is strictly greater than zero at the steady state. The temporization of a net system is given by the vector of firing rates assigned to the transitions ($\boldsymbol{\lambda}$). The addition of time with an appropriate $\boldsymbol{\lambda}$ may cause a non live untimed system to reach a live steady state. In Chapter 4 the set of $\boldsymbol{\lambda}$ that allows the net to be structurally live is characterized for mono-T-semiflow nets under infinite servers semantics. The shape of such a set highly depends on the conflicts of the net. If the dimension of the set is less than the number of transitions, any variation of the vector $\boldsymbol{\lambda}$ may cause the new $\boldsymbol{\lambda}$' to be out of the set, thus, entailing the impossibility of reaching a live steady state. From this idea the concept of *critical liveness* is derived. On the opposite side to critical liveness, *robust liveness* applies to those systems for which a live steady state can be reached for any vector $\boldsymbol{\lambda}$. A strong topological condition establishes which systems are robust live: Every transition t must have an input place whose only output transition is t. If this condition holds the system is structurally live for any $\boldsymbol{\lambda}$, otherwise there exists at least a $\boldsymbol{\lambda}$ such that the system deadlocks for any initial marking.

The *marking evolution* of a timed net system is always contained in the set of reachable markings of the system seen as untimed. This becomes clear if one thinks that the trajectory of the timed system (under any time interpretation) can be imitated by the untimed system by appropriately firing its transitions. This way, if the timed system can reach a deadlock marking, the untimed system can reach as well the same deadlock marking. In other words, liveness of the timed system is a necessary condition for liveness of the untimed system. This relationship between untimed and timed systems allows one to improve the liveness conditions for untimed systems obtained previously. More specifically, the results on robust liveness for timed systems yield a strong necessary topological condition for liveness of untimed systems (Theorem 4.14).

The *performance* of a timed net system can be measured as the throughput of the transitions in the steady state. In a mono-T-semiflow system, the throughput of the transitions in the steady state is proportional to the only T-semiflow of the net. Hence, once the throughput of one transition is known, the throughput of the rest of transitions can be immediately computed. Under infinite servers semantics the flow of one transition is proportional to the marking of one of its input places. In order to determine which of the input places can define the flow of a transition in the steady state a Branch & Bound algorithm has been developed in Chapter 5. The width, and therefore the number of nodes, of the tree strongly depends on the number of input places of the transitions. The mentioned algorithm can be used both for the

computation of upper and lower throughput bounds. Clearly, if both bounds are the same, the algorithm is yielding the exact throughput of the system in the steady state.

A detailed a priori study of the mono-T-semiflow net structure usually helps to reduce considerably the size of the tree in the Branch & Bound algorithm: By only considering the structure, it is possible to establish which of the input places of a given transition can really define the flow of the transition in the steady state. Furthermore, by removing some constraints, the Branch & Bound algorithm can be relaxed into a linear programming problem to compute upper throughput bounds. The goal of such programming problem is searching for the bottleneck P-semiflow of the net. Although, the upper bound yielded by the linear programming problem is in general less tight than the one of the Branch & Bound algorithm, it has the advantage that it can be computed in polynomial time.

Controllability and *observability* are dual concepts in systems theory. On the one hand, controllability applies to those systems whose state can be driven anywhere in the state space by the input actions of an external system. On the other hand, observability stands for those systems whose state can be completely estimated by an external observer. Both properties have been thoroughly studied in the literature in the framework of time invariant linear systems. Some of those results can be applied and improved if focusing on timed continuous Petri nets.

With respect to observability, the concept of *structural observability* has been defined and developed in Chapter 6. A place is said to be structurally observable if its marking can be estimated for any vector of firing rates of the transitions. In order to compute the set of structurally observable places a fix point algorithm has been designed. The observability matrix of linear systems can also be used for continuous Petri nets: One observability matrix can be computed per linear system (PT-set) of the Petri net. Each observability matrix is associated to an equation that yields an estimate for the net marking. Fortunately, some rules have been developed to filter those observability matrices that with certainty are not yielding a right estimate for the net marking. Such estimates have been classified in three groups: *infeasible*, *non-coherent* and *suspicious* estimates. Observers have been designed by considering one Luemberger observer per linear system of the Petri net. The estimates of the observers present many similarities to the estimates computed algebraically. Non appropriate estimates of the observers are identified and filtered. The resulting observer is an algorithm that combines the non filtered estimates and a simulation of the system. The estimate given by this observer does not change sharply when the PT-set of the system commutes. Under some conditions, the use of the simulation offers the possibility of estimating the non observable subspace of the system during a given time period.

The study of controllability in hybrid systems is nowadays an issue of increasing interest. The non linear evolution of hybrid systems poses serious problems when

trying to establish simple to check controllability conditions and easy to design controllers. Regarding to discrete time systems, two important parameters that have to be carefully considered are accuracy and computational time to obtain a control law. Usually, there exists a tradeoff between these two parameters. In order to increase the accuracy, the length of the sampling period is usually taken short. This makes that the number of steps required for a given time horizon is high. A large number of steps often involves a high computational load to obtain the control law. A control method has been presented in Chapter 7 for timed continuous nets under finite servers semantics. The method allows one to get rid of the tradeoff between accuracy and computational time, transforming a given timed net system into an event-based Mixed Logical Dynamical System (eMLD). The main feature of the obtained eMLD is that the occurrence of an event exactly coincides with the change of the linear system driving the evolution of the Petri net (a place becomes empty). By using eMLDs, optimal control problems over a given horizon can be solved as mixed integer linear programming problems.

This research work still presents many open directions to be considered and investigated. Chapter 8 shows how to take profit of the developed concepts and results through examples modelled with finite and infinite servers semantics. It also makes clear the interest of introducing new time interpretations for "realistic" modelling. Such new interpretations need the development of a theory that analyzes them.

Bibliography

[AD98a] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8(1):159–188, 1998.

[AD98b] H. Alla and R. David. A modeling and analysis tool for discrete event systems: Continuous Petri net. *Performance Evaluation*, 33:175–199, 1998.

[BBDSV02] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. L. Sangiovanni-Vincentelli. Design of observers for hybrid systems. In Claire J. Tomlin and Mark R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, pages 76–89. Springer-Verlag, Berlin Heidelberg New York, 2002.

[BF87] An Introduction to State-Space Methods B. Friedland. *Control system design*. McGraw-Hill, 1987.

[BGM00] F. Balduzzi, A. Giua, and G. Menga. First-order hybrid Petri nets: a model for optimization and control. *IEEE Trans. on Robotics and Automation*, 16(4):382–399, 2000.

[BM99] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[Bra83] G. W. Brams. *Réseaux de Petri: Théorie et Pratique*. Masson, 1983.

[CAC+93] G. Chiola, C. Anglano, J. Campos, J. M. Colom, and M. Silva. Operational analysis of timed Petri nets and application to the computation of performance bounds. In *Proceedings of the 5th International Workshop on Petri Nets and Performance Models*, pages 128–137, Toulouse, France, October 1993. IEEE-Computer Society Press.

[CCS91] 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.

[CGSJ03] D. Corona, A. Giua, C. Seatzu, and J. Júlvez. Observers for nondeterministic λ-free labelled Petri nets. In *Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages –, Lisbon, Portugal, September 2003.

[CHEP71] F. Commoner, A. W. Holt, S. Even, and A. Pnueli. Marked directed graphs. *Journal on Computer Systems Science*, 5:72–79, 1971.

[CNT99] G. Ciardo, D. Nicol, and K. S. Trivedi. Discrete-event simulation of fluid stochastic Petri nets. *IEEE Trans. on Software Engineering*, 25(2):207–217, 1999.

[CS92] 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.

[DA87] R. David and H. Alla. Continuous Petri nets. In *Proc. of the 8th European Workshop on Application and Theory of Petri Nets*, pages 275–294, Zaragoza, Spain, 1987.

[Dag95] C. Daganzo. A finite difference approximation of the kinematic wave model of traffic flow. *Transportation Research B*, 29B(4):261–276, 1995.

[DHP+93] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993.

[ECS93] J. Ezpeleta, J. M. Couvreur, and M. Silva. A new technique for finding a generating family of siphons, traps and ST-components. application to coloured Petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1993*, volume 674 of *Lecture Notes in Computer Science*, pages 126–147. Springer, 1993.

[Flo95] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, 1995.

[GMD03] Jorge M. Goncalves, Alexandre Megretski, and Munther A. Dahleh. Global analysis of piecewise linear systems using impact maps and surface lyapunov functions. *IEEE Trans. on Automatic Control*, 48(12):2089–2106, 2003.

[GS02] A. Giua and C. Seatzu. Observability of place/transition nets. *IEEE Transactions on Automatic Control*, 47(9):1424–1437, September 2002.

[GSJar] A. Giua, C. Seatzu, and J. Júlvez. Marking estimation of Petri nets with pairs of nondeterministic transitions. *Asian Journal of Control, special issue on "Control of Discrete Event Systems"*, To appear.

[Hac72] M. H. T. Hack. Analysis of production schemata by Petri nets. Master's thesis, M.I.T., Cambridge, MA, USA, 1972. (Corrections in *Computation Structures Note* 17, 1974).

[HB01] S. Hoogendoorn and P. Bovy. State-of-the-art of vehicular traffic flow modelling. *Special Issue on Road Traffic Modelling and Control of the Journal of Systems and Control Eng. Proc. of the IME I*, 2001.

[Hel97] D. Helbing. Traffic data and their implications for consistent traffic flow modelling. In M. Papageorgiou and A. Pouliezos, editors, *Transportation Systems (IFAC, Chania, Greece)*, volume 2, pages 809–814, 1997.

[JB05] J. Júlvez and R. Boel. Modelling and controlling traffic behaviour with continuous Petri nets. In *Proceedings of the* 16^{th} *triennial world congress of the International Federation of Automatic Control (IFAC 2005)*, 2005. Submitted.

[JBRS04] J. Júlvez, A. Bemporad, L. Recalde, and M. Silva. Event-driven optimal control of continuous Petri nets. In *43rd IEEE Conference on Decision and Control (CDC)*, Paradise Island. Bahamas, 2004.

[JJRS04a] E. Jiménez, J. Júlvez, L. Recalde, and M. Silva. Relaxed continuous views of discrete event systems: Petri nets, Forrester diagrams and ODES. In *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, The Hague, The Netherlands, October 2004.

[JJRS04b] J. Júlvez, E. Jiménez, L. Recalde, and M. Silva. Design of observers for timed continuous Petri net systems. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, The Hague, The Netherlands, 2004.

[JJRS04c] J. Júlvez, E. Jiménez, L. Recalde, and M. Silva. On observability in timed continuous Petri net systems. In *First International Conference on the Quantitative Evaluation of Systems (QEST). IEEE Computer Society.*, Enschede, The Netherlands., 2004.

[Joh99] M. Johansson. *Piecewise Linear Control Systems.* PhD thesis, Lund Institute of Technology, 1999.

[JRS] J. Júlvez, L. Recalde, and M. Silva. Deadlock-freeness analysis of continuous mono-T-semiflow Petri nets. *IEEE Transactions on Automatic Control*. Submitted.

[JRS02] J. Júlvez, L. Recalde, and M. Silva. On deadlock-freeness analysis of autonomous and timed continuous mono-T-semiflow nets. In *Proceedings of the* 41^{st} *IEEE Conference on Decission and Control (CDC 2002)*, pages 781–786, Las Vegas, USA, December 2002.

[JRS03] 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.

[JRS04] J. Júlvez, L. Recalde, and M. Silva. Steady state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 2004. Accepted for publication.

[KPD+02] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavis, and F. Middelham. Traffic flow modelling of large-scale motorway using the macroscopic modeling tool metanet. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):282–292, 2002.

[Lue71] D.G. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602, December 1971.

[Mig02] D. Mignone. *Control and Estimation of Hybrid Systems with Mathematical Optimization*. PhD thesis, Automatic Control Laboratory-ETH, Zurich, 2002.

[Mur83] K. G. Murty. *Linear Programming*. Wiley and Sons, 1983.

[Mur89] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[Oga95] K. Ogata. *Discrete-Time Control Systems, 2nd. ed.* Prentice Hall, 1995.

[Pet81] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.

[Rec98] L. Recalde. *Structural Methods for the Design and Analysis of Concurrent Systems Modeled with Place/Transition Nets*. PhD thesis, DIIS. Univ. Zaragoza, July 1998.

[RJS02] L. Recalde, J. Júlvez, and M. Silva. Steady state performance evaluation for some continuous Petri nets. In *Proceedings of the* 15^{th} *triennial world congress of the International Federation of Automatic Control (IFAC 2002)*, page N 479, Barcelona, Spain, July 2002.

[RS00] L. Recalde and M. Silva. PN fluidification revisited: Semantics and steady state. In J. Zaytoon S. Engell, S. Kowalewski, editor, *ADPM 2000: 4th Int. Conf. on Automation of Mixed Processes: Hybrid Dynamic Systems*, pages 279–286, 2000.

[RS01] L. Recalde and M. Silva. Petri Nets fluidification revisited: Semantics and steady state. *APII-JESA*, 35(4):435–449, 2001.

[RTRRLM03] A. Ramirez-Trevino, I. Rivera-Rangel, and E. Lopez-Mellado. Observability of discrete event systems modeled by interpreted Petri nets. *IEEE Trans. on Robotics and Automation*, 19(4):557–565, 2003.

[RTS98] 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.

[RTS99] 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.

[SC88] M. Silva and J. M. Colom. On the computation of structural synchronic invariants in P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *Lecture Notes in Computer Science*, pages 387–417. Springer, 1988.

[SC95] M. Silva and J. Campos. Structural performance analysis of stochastic Petri nets. In *IEEE IPDS '95*, pages 61–70. IEEE Computer Society Press, 1995.

[SGL02] Z. Sun, S.S. Ge, and T.H. Lee. Controllability and reachability criteria for switched linear systems. *Automatica*, 38:775–786, 2002.

[Sif78] J. Sifakis. Structural properties of Petri nets. In J. Winkowski, editor, *Mathematical Foundations of Computer Science 1978*, pages 474–483. Springer, 1978.

[Sil85] M. Silva. *Las Redes de Petri: en la Automática y la Informática*. AC, 1985.

[Sil93] M. Silva. Introducing Petri nets. In *Practice of Petri Nets in Manufacturing* [DHP+93], pages 1–62.

[Son81] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, April 1981.

[SR02] M. Silva and L. Recalde. Petri nets and integrality relaxations: A view of continuous Petri net models. *IEEE Trans. on Systems, Man, and Cybernetics*, 32(4):314–327, 2002.

[SR04] M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 2004. To appear.

[STC98] 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.

[SvdB01] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[TB04] F.D. Torrisi and A. Bemporad. HYSDEL — A tool for generating computational hybrid models. *IEEE Trans. Contr. Systems Technology*, 12(2), March 2004. `http://control.ethz.ch/~hybrid/hysdel`.

[TCS97] 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.

[Ter94] E. Teruel. *Structure Theory of Weighted Place/Transition Net Systems: The Equal Conflict Hiatus*. PhD thesis, DIEI. Univ. Zaragoza, June 1994.

[TK93] K. Trivedi and V. G. Kulkarni. FSPNs: Fluid stochastic Petri nets. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *Lecture Notes in Computer Science*, pages 24–31. Springer, 1993.

[TS96] E. Teruel and M. Silva. Structure theory of equal conflict systems. *Theoretical Computer Science*, 153(1-2):271–300, 1996.

[VCS02] R. Vidal, A. Chiuso, and S. Soatto. Observability and identifiability of jump linear systems. In *Proceedings of the* 41^{st} *IEEE Conference on Decission and Control (CDC 2002)*, pages 3614–3619, Las Vegas, USA, December 2002.

[VCSS03] R. Vidal, A. Chiuso, S. Soatto, and S. Sastry. Observability of linear hybrid systems. In *Hybrid Systems: Computation and Control*, volume 2623 of *Lecture Notes in Computer Science*, pages 526–539. Springer-Verlag, 2003.

[ZRS01] A. Zimmermann, D. Rodríguez, and M. Silva. A two phase optimisation method for Petri net models of manufacturing systems. *Journal of Intelligent Manufacturing*, 12(5):421–432, October 2001.