Discrete Event Dynamic Systems: Theory and Applications, 11, 9–40, 2001.

On Hybrid Petri Nets

RENÉ DAVID
Laboratoire d'Automatique de Grenoble (INPG-UJF-CNRS), B.P. 46, 38402 Saint-Martin-d'Hères, France

HASSANE ALLA
Laboratoire d'Automatique de Grenoble (INPG-UJF-CNRS), B.P. 46, 38402 Saint-Martin-d'Hères, France

Abstract. Petri nets (PNs) are widely used to model discrete event dynamic systems (computer systems, manufacturing systems, communication systems, etc). Continuous Petri nets (in which the markings are real numbers and the transition firings are continuous) were defined more recently; such a PN may model a continuous system or approximate a discrete system. A hybrid Petri net can be obtained if one part is discrete and another part is continuous.

This paper is basically a survey of the work of the authors' team on hybrid PNs (definition, properties, modeling). In addition, it contains new material such as the definition of extended hybrid PNs and several applications, explanations and comments about the timings in Petri nets, more on the conflict resolution in hybrid PNs, and connection between hybrid PNs and hybrid automata.

The paper is illustrated by many examples.

Keywords: Petri nets, hybrid, extended, continuous, modeling

1. Introduction

Many systems are naturally hybrid, i.e., their modeling needs at least one continuous state variable and at least one discrete state variable (more information on the definition of a hybrid system can be found in (David, 1997)). In some cases, a discrete system,[1] or part a of system, can be approximated by a continuous model (Gershwin and Schick, 1980; Dubois and Forestier, 1982) and this approximation may be very good (David *et al.*, 1990; Mandelbaum and Chen, 1991).

Petri nets (PNs) are widely used to model discrete systems (computer systems, manufacturing systems, communication systems, etc). In a PN, the marking of a place may correspond either to the Boolean state of a device (for example a resource is available or not), or to an integer (for example the number of parts in a buffer). A general analysis method is to compute the set of reachable states and deduce the different properties of the system. But when a PN contains a large number of tokens, the number of reachable states explodes and this is a practical limitation of the use of Petri nets. To illustrate this point, consider a manufacturing line composed of three machines M_1, M_2 and M_3 in order, and two intermediate buffers B_1 and B_2 with respective finite capacities, C_1 and C_2. The parts move on the machines, and wait in the intermediate buffers if required. We assume that there are always unworked parts upstream M_1 and available space downstream M_3. The number of reachable states of this system is $N = 2^3(C_1 + 1)(C_2 + 1)$; then $N = 1\,352$ for $C_1 = C_2 = 12$. For a set composed of 10 machines and 9 buffers each with capacity 12, $N = 2^{10} \times 13^9$ which is greater than 10^{13} states!

This observation led us to define continuous PNs and hybrid PNs. In a continuous PN, the markings of places are real numbers and the firing of transitions is a continuous process. For the example considered, the flow of parts on the machine may be approximated by a continuous flow and the numbers of parts in the buffers may be approximated by real numbers. However, the state of each machine (operational or not) is necessarily discrete. Hence, a hybrid model can be used for this system (presented in Section 3, Fig. 14).

Continuous Petri nets were introduced in (David and Alla, 1987). The concept of hybrid Petri net, introduced in the same paper, was developed in (Le Bail *et al.*, 1991).

A continuous PN may be either autonomous (no time is involved, formal definition in (David and Alla, 1990)) or with firing speeds associated with transitions. A timed model may be used for the performance evaluation of systems. Various timed continuous PN models have been defined which differ by the calculation of the instantaneous firing speeds of the transitions (David and Alla, 1987, 1990) (Le Bail *et al.*, 1993; Dubois *et al.*, 1994). They provide good approximations for performance evaluation when a PN contains a large number of tokens. All the models mentioned above work on the same basic rule (Equation (12) in Section 3.3.4). The only difference is the way in which the instantaneous firing speeds are defined; it follows that other definitions of this firing speed can be chosen (see Section 3.5.2). Other authors have added some timings to places in continuous Petri nets (Brinkman and Blaauboer, 1990). In (Olsder, 1993), the continuous flows are studied with the help of Max-plus algebra. Various theoretical results on continuous timed Petri nets, including a correspondence between these nets and a Markov decision process are presented in (Cohen *et al.*, 1995, 1998). The modeling power of hybrid PNs and a comparison with Bond graphs is presented in (Pettersson and Lennartson, 1995).

In the fluid stochastic PNs proposed in (Trivedi and Kulkani, 1993), the arcs represent fluid flows. In this paper, the authors model the same kind of systems as the hybrid PNs with a stochastic discrete part (Section 3.5.3), while other simulation possibilities are added in (Ciardo *et al.*, 1997). Some authors have explicitly added new concepts and results to our initial definition of hybrid Petri nets: special places and transitions have been added in order to model systems processing batches of parts (Demongodin, Prunet, *et al.*, 1992, 1998); differential PNs are an extension capable of modeling hybrid systems whose continuous part is represented by differential equations (Demongodin and Koussoulas, 1998); in (Balduzzi *et al.*, 1998), the authors use hybrid stochastic Petri nets, in which the discrete transitions may be either immediate or stochastic or deterministically timed, in order to model flexible manufacturing systems; hybrid high-level nets were introduced and used for modeling and simulation in (Weiting, 1996).

In this paper, the authors present both a survey of their previous results on hybrid Petri nets, including a part of (Alla and David, 1998), and new material, particularly a definition and applications of *extended hybrid Petri nets*, specification of *priority rules between discrete and continuous parts*, and a connection between *hybrid Petri nets and hybrid automata*.

The paper is organized as follows. *Autonomous* hybrid PNs are presented in Section 2, *timed* hybrid PNs in Section 3, and application examples in Section 4. Section 5 describes the move from a hybrid PN to a hybrid automaton. Section 6 is the conclusion.

2. Automomous Hybrid Petri Nets

The models described in this section are autonomous, i.e., time is not involved. An *autonomous* Petri net enables a *qualitative* study of all possible behaviors. The word "autonomous" may be implicit when not specified.

2.1. Intuitive Presentation

It is assumed that the reader is familiar with Petri nets (Peterson, 1981; Murata, 1989; David and Alla, 1992).

Figure 1.a illustrates a Petri net. Let us note m_i the number of tokens (or marks[2]) in place P_i. The marking of the PN in Fig. 1.a is (2, 1, 0, 0), which corresponds to the increasing order of indexes,[3] i.e., $\mathbf{m} = (m_1, m_2, m_3, m_4)$. The transitions T_1 and T_3 are enabled since there is at least one token in each input place of these transitions. Firing consists of removing a token from each input place and adding a token to each output place of the transition fired. The firing of T_1 would lead to the marking $(1, 1, 1, 0)$, and the firing of T_3 would lead to $(2, 0, 0, 1)$. All the possible firings appear on the marking graph in Fig. 1.b. Note that there are two marking invariants: $m_1 + m_3 = 2$ and $m_2 + m_4 = 1$. The state of the PN can thus be represented by (m_1, m_2) instead of (m_1, m_2, m_3, m_4) which is redundant. This enables us to represent the marking graph in the plane: see Fig. 1.c. Six possible states can be observed.

Remark 1 In a discrete PN, from a marking $\mathbf{m}$, a *firing sequence* implies a string of successive markings. The *characteristic vector* $\mathbf{s}$ of a firing sequence S is a vector for which each component is an integer corresponding to the number of firings of the corresponding transition. Then a marking $\mathbf{m}$ reached from $\mathbf{m}_0$ by firing of a sequence S can be deduced using the *fundamental relation*:

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{W} \cdot \mathbf{s}, \tag{1}$$

where $\mathbf{W}$ is the incidence matrix.

2.1.2. Autonomous Continuous Petri Nets

In (David and Alla, 1990), autonomous continuous PNs are defined as a limit of a discrete PN: a mark is split into k tokens, and k tends to infinity. Figure 1.d shows a continuous PN (the continuous places and transitions are represented by a double line). The initial marking shown is also $(2, 1, 0, 0)$ but in this case the markings are real numbers and no longer integers. In this state the transitions T_1 and T_3 are enabled, i.e., firable, since the markings of their input places are not nil. Transition T_1 can be fired, for example, but we now define a "*firing quantity*" which is a real number taken from the continuous interval $[0, 1]$; the maximum value, 1 in this case, corresponds to m_2 (which is the minimum of m_1 and m_2). For a firing quantity 0.2, the marking $(1.8, 1, 0.2, 0)$ is obtained. As above, the marking invariants can

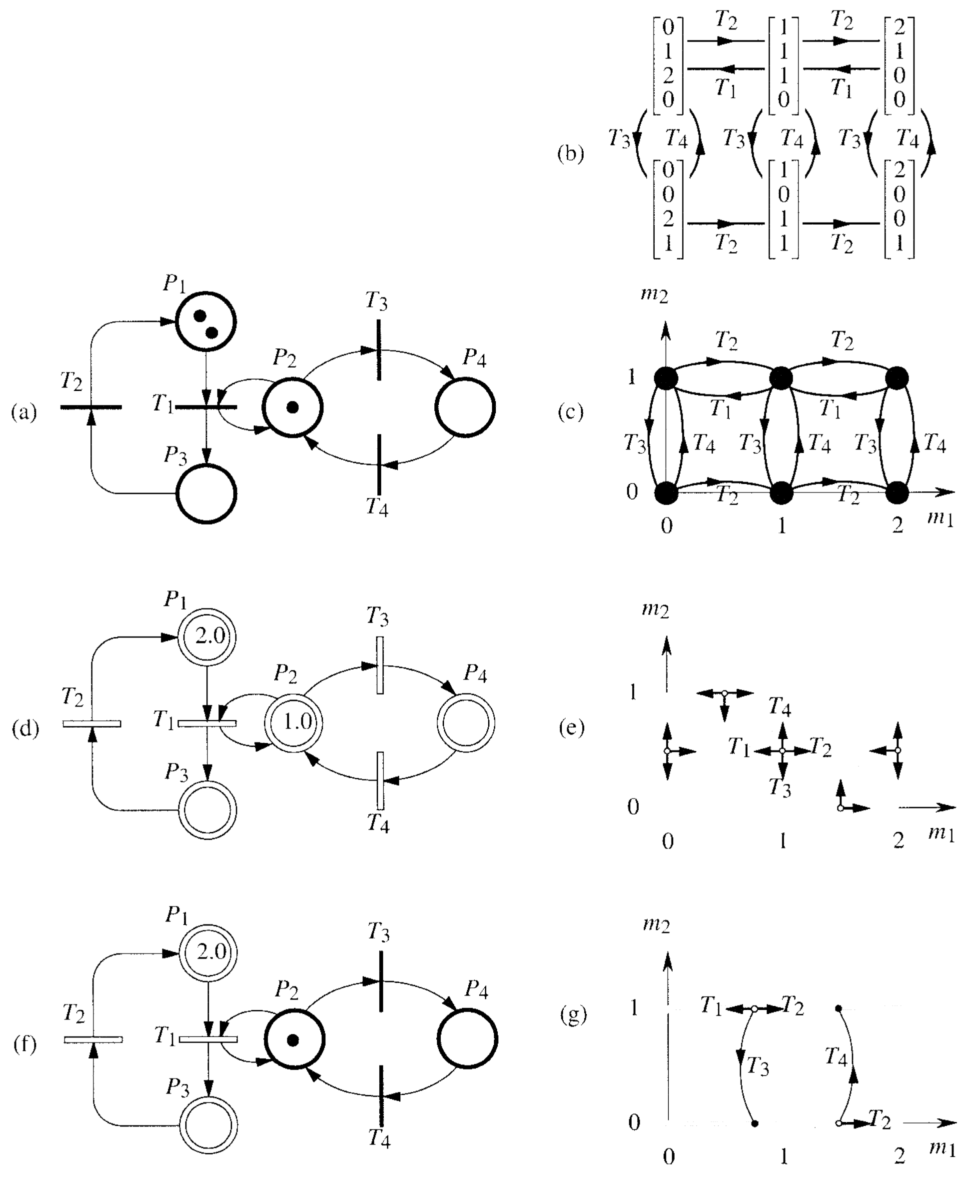

Figure 1. (a)(b)(c) Discrete Petri net. (d)(e) Continuous Petri net. (f)(g) Hybrid Petri net.

be used to represent the state of the system in the plane (m_1, m_2): see Fig. 1.e. We observe that there is an infinite number of accessible markings corresponding to the shaded part of the plane. Arrows mark the transitions that can be fired at different points of the plane.

Remark 2

a) In a continuous PN, from a marking $\mathbf{m}$, a firing sequence S implies a trajectory corresponding to a string of successive markings. The *characteristic vector* $\mathbf{s}$ of a trajectory is a vector for which each component is a real number corresponding to a firing quantity of the corresponding transition. Then a marking $\mathbf{m}$ reached from $\mathbf{m}_0$ by firing of a sequence S can be deduced using the *fundamental relation*: $\mathbf{m} = \mathbf{m}_0 + \mathbf{W} \cdot \mathbf{s}$.

The fundamental relation of a continuous PN is identical to the fundamental relation of a discrete PN. Then, every property of a discrete PN resulting from this relation can be transposed to continuous PNs. In particular, the results for P-invariants and T-invariants are similar for a continuous PN and for a discrete PN.

b) The concepts of liveness, boundedness and deadlock-freeness, allowing some logical properties to be studied, are quite similar for continuous and discrete Petri nets.

2.1.3. Autonomous Hybrid Petri Nets

Figure 1.f shows a hybrid PN. The continuous places are P_1 and P_3, the continuous transitions are T_1 and T_2, the discrete places P_2 and P_4, and the discrete transitions T_3 and T_4. The transitions T_1 and T_3 are enabled and thus firable. Let us consider the firing of the continuous transition T_1. For a firing quantity 0.1, the marking $(1.9, 1, 0.1, 0)$ is obtained. A marking quantity 0.1 has been removed from P_1 and P_2 which are the input transitions, and the same quantity has been added to P_3 and P_2 which are the output transitions. We observe that the marking in the discrete place P_2 is still an integer (since the same quantity has been removed and added). Figure 1.g shows that the reachable markings are the two shaded segments. We move continuously along a segment by firing T_1 or T_2. We move from one segment to another for the discrete firing of T_3 or T_4.

In the 3 cases described in Fig. 1, when $m_2 = 0$, the transition T_1 is not enabled. In Fig. 1.c there is no arc marked T_1 when $m_2 = 0$. In Fig. 1.e there is no arrow in the direction of T_1 when $m_2 = 0$. Likewise in Fig. 1.g we cannot move on the segment corresponding to $m_2 = 0$ by firing T_1.

2.2. Hybrid Petri Nets

In Section 2.2.1, an autonomous hybrid PN is formally defined. Then, in Section 2.2.2, the behaviors which can be modeled are illustrated.

2.2.1. Definition

Definition 1 An **autonomous hybrid PN** is a sextuple $Q = \langle P, T, Pre, Post, \mathbf{m}_0, h\rangle$ such that:

$P = \{P_1, P_2, \ldots, P_n\}$ is a finite, not empty, set of places;
$T = \{T_1, T_2, \ldots, T_m\}$ is a finite, not empty, set of transitions;
$P \cap T = \varnothing$, i.e. The sets P and T are disjointed;

h: $P \cap T \to \{D, C\}$, called “hybrid function,” indicates for every node whether it is a discrete node (sets P^D and T^D) or a continuous node (sets P^C and T^C);
Pre: $P \times T \to \mathcal{R}^+$ or $\mathcal{N}$, is the input incidence mapping;[4]
Post: $P \times T \to \mathcal{R}^+$ or $\mathcal{N}$, is the output incidence mapping;
$\mathbf{m}_0$: $P \to \mathcal{R}^+$ or $\mathcal{N}$ is the initial marking.
In the definition of *Pre*, *Post* and $\mathbf{m}_0$, $\mathcal{N}$ corresponds to the case where $P_i \in P^D$, and $\mathcal{R}^+$ corresponds to the case where $P_i \in P^C$.

Pre and Post functions must meet the following criterion: if P_i and T_j are a place and a transition such that $P_i \in P^D$ and $T_j \in T^C$ then $Pre(P_i, T_j) = Post(P_i, T_j)$ must be verified. ■

As usual, ${}^\circ P_i$ and P_i° denote respectively the sets of input transitions and the set of output transitions of place P_i; ${}^\circ T_j$ and $T_j{}^\circ$ denote respectively the sets of input places and the set of output places of transition T_j. To abbreviate, D-place, D-transition, C-place, and C-transition, may stand for discrete place, discrete transition, continuous place, and continuous transition, respectively.

In a hybrid PN, the *characteristic vector* $\mathbf{s}$ of a sequence S is a vector for which each component is either an integer corresponding to the number of firings of a D-transition or a non-negative real number corresponding to a firing quantity of a C-transition. A marking $\mathbf{m}$ can be deduced from a marking $\mathbf{m}_0$ due to a sequence S, using the fundamental relation:

$$\mathbf{m} = \mathbf{m}_0 + W \cdot \mathbf{s}. \tag{1'}$$

In Equation (1), all the components of the vector s are integers, while in (1′) the components of $\mathbf{s}$ are either integer or non-negative real numbers: this is the only difference between both equations. Then, we have the same properties as stated in Remark 2, i.e., the results for P-invariants and T-invariants are similar for a hybrid PN and for a discrete PN.

2.2.2. A hybrid Petri net allows modeling of . . .

Informally, there are two parts in a hybrid PN, a discrete part (containing P^D and T^D) and a continuous part (containing P^C and T^C), and these parts are interconnected thanks to arcs linking a discrete node (place or transition) to a continuous node (transition or place). In some cases, one part can influence the behavior of the other part without changing its own marking. In other cases, the firing of a D-transition can modify both the discrete and the continuous marking. Here are some illustrating examples.

a) *Influence of the discrete part on the continuous part*.

Figure 1.f is an illustration of this influence. Let us assume that the C-places P_1 and P_3 correspond to tanks between which a liquid flows. The marking of a place corresponds to the quantity of liquid in the corresponding tank. The transition T_1 represents pumping (moving from P_1 into P_3). The transition T_2 corresponds to a gravitational flow. When the pump is running (a token in the D-place P_2), the continuous part evolves by continuous

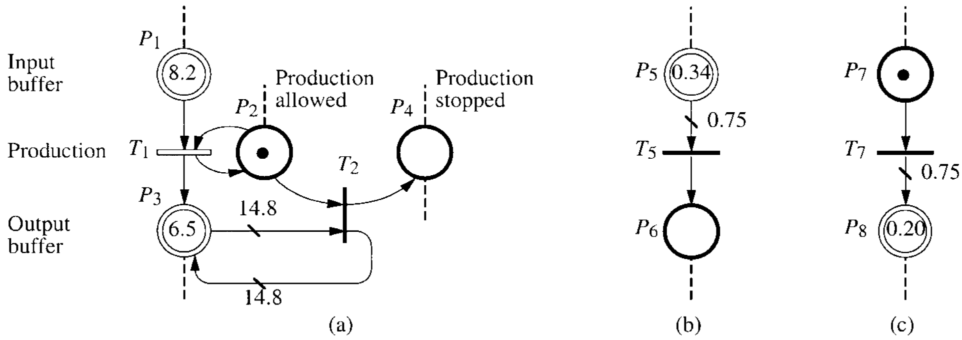

Figure 2. (a) Influence of the continuous part on the discrete part (and vice-versa). (b) and (c) Transformation of continuous marking into discrete marking and vice-versa.

firing of the transitions T_1 and T_2. When the pump is stopped (a token in P_4 but not in P_2), the transition T_1 is no longer enabled and is thus no longer fired. In the continuous part, only the transition T_2 can still be fired as long as $m_3 > 0$.

b) *Influence of the continuous part on the discrete part.*
An example is given in Fig. 2.a. In this case the continuous part represents a production system. The transition T_1 corresponds to the production of a machine, continuous production or approximation by a continuous flow of a discrete production. When the output buffer reaches a certain level, 14.8 in Fig. 2.a, production stops (firing of T_2). This transition takes priority over the continuous transitions (see Section 3.4).

c) *Converting a continuous marking into a discrete marking, and vice-versa.*
Figure 2.b illustrates the conversion of a continuous marking into a discrete marking by the firing of a D-transition. In this figure, the transition T_5 is not enabled because $m_5 < 0.75$, i.e., the marking of P_5 is less than the weight of the arc $P_5 \rightarrow T_5$. Figure 2.c illustrates the opposite conversion, i.e., converting a discrete marking into a continuous marking. Transition T_7 is enabled; firing consists of removing a token (integer) from place P_7 and adding a marking quantity 0.75 to P_8.

In the general case, a *discrete transition* may have input and output places, either continuous or discrete, without restriction. A *continuous transition* can also have discrete and continuous input places as well as discrete and continuous output places. However *all discrete input places must also be output places, and vice-versa* (i.e., output places must be input places), *with arcs of the same weight.* This is illustrated in Fig. 1.f and 2.a, for example. This is a vital property for preserving the integral character of discrete marking (it follows that the *firing of a continuous transition cannot modify the marking of the discrete part*).

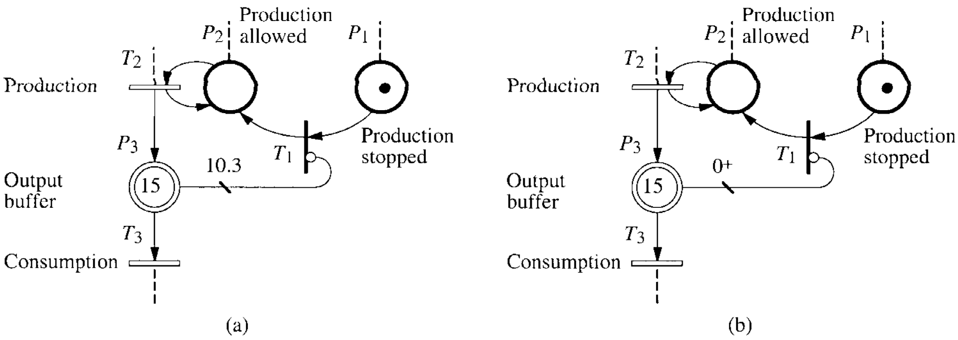

Figure 3. (a) Threshold test on a continuous place. (b) Zero test on a continuous place.

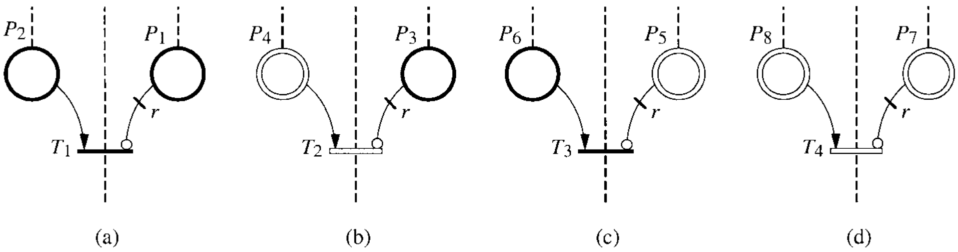

Figure 4. Inhibitor arc in a hybrid PN. (a) and (b) Weight $r \in \mathcal{N}$. (c) and (d) Weight $r \in \mathcal{R}^+ \cup \{0^+\}$.

2.3. Extended Hybrid Petri Nets

In a discrete extended PN, an inhibitor arc of weight r from a place P_i to a transition T_j allows the firing of T_j only if the marking of P_i is less than r. The same concept may be used in a hybrid PN. In Fig. 3.a, the inhibitor arc from P_3 to T_1 has a weight $r = 10.3$. This is a threshold test. It means that the transition T_1 cannot be fired, i.e., production is stopped, as long as the level of products in the output buffer modeled by P_3 is at least 10.3. As soon as the consumption (corresponding to the continuous firing of T_3) has lowered the marking in P_3 such that $m_3 < r = 10.3$, the transition T_1 can be fired since there is a token in P_1. This firing adds a token to P_2 and the production (continuous firing of T_2) can start again.

If the inhibitor arc has its origin at a discrete place (in a discrete or a hybrid PN) and has a weight $r = 1$, it corresponds to a zero test. As a matter of fact, the corresponding transition can be fired only if $m_i < 1$, i.e. if $m_i = 0$ since m_i is an integer. Now, how can we model a zero test if the origin place is continuous? In order to be able to use the concept of zero test, we introduce here[5] the following convention: 0^+ *represents a weight infinitely small but not nil*. For example, in Fig. 3.b, the transition T_1 can be enabled only if $m_3 = 0$.

All the cases of inhibitors arcs encountered in a hybrid PN are presented in Fig. 4. There are 4 cases of firing of transitions in case of inhibitor arcs (it is assumed that other input places can exist, according to Definition 1). The weight r of the inhibitor arc is an integer if the corresponding place is discrete (P_1 in Fig. 4.a and P_3 in Fig. 4.b), it is a real positive number or the conventional value 0^+ if the corresponding place is continuous (P_5 in Fig. 4.c and P_7 in Fig. 4.d).

Let us now define an extended Petri net.

Definition 2 The definition of an **extended hybrid PN** (autonomous) is similar to the definition of a hybrid PN (Def. 1), except that:

1) one can have, in addition, inhibitor arcs (according to the previous explanation);

2) the weight of an arc (inhibitor or ordinary) whose origin is a continuous place takes its value in $\mathcal{R}^+ \cup \{0^+\}$ instead of $\mathcal{R}^+$;

3) the marking of a continuous place takes its value in $\mathcal{R}^+ \cup \{0^+\}$ instead of $\mathcal{R}^+$. ■

The use of the conventional value 0^+ may also have other applications than the case of zero test. Two examples will be given in Section 4, after the timed hybrid PNs have been presented.

3. Timed Hybrid Petri Nets

Some basic concepts about the time in discrete PNs are recalled in Section 3.1 and modeling of time in hybrid PNs is presented in Section 3.2. Then, a hybrid PN in which the timings (firing speeds for C-transitions and delays for D-transitions) are constant, is given in Section 3.3. The conflicts in a hybrid PN are analyzed in Section 3.4, and cases where the timings are not constant are presented in Section 3.5.

3.1. General Information on Time in Discrete PNs

Two basic models of timed discrete PNs have been defined; time is associated either with the places (Sifakis, 1977) or with the transitions (Ramchandani, 1973). It is well known that transfers are possible from one model to another. In a PN, it is natural to associate with a place a state which has some duration and to associate with a transition a change of state, this change having no duration. It is then natural to associate the duration of some operation or state with a place, and the time of waiting for an event to the transition which is fired when the event occurs. Let us illustrate these ideas with examples.

Figure 5.a represents a system made of two machines M_A and M_B on which four customers pass alternatively. Machine M_A has a single server, while M_B is a double-server machine (i.e., two customers can be processed at the same time). The processing time of M_A is d_A, and the processing time of both servers of M_B is d_B. In Fig. 5.a, the state of the system is as follows: the tokens in P_1 represent customers waiting for an operation on the machine M_A; a customer is processed by the machine M_A (token in P_2), hence the machine is not available (no token in P_2') and the transition T_1 cannot be fired. A customer is processed by M_B (token in P_4); a server of this machine remains idle (token in P_4') since there is no token in P_3. Operation times d_A and d_B are naturally associated with the places P_2 and P_4; this means that a token arriving in P_2 must remain during d_A before allowing firing of

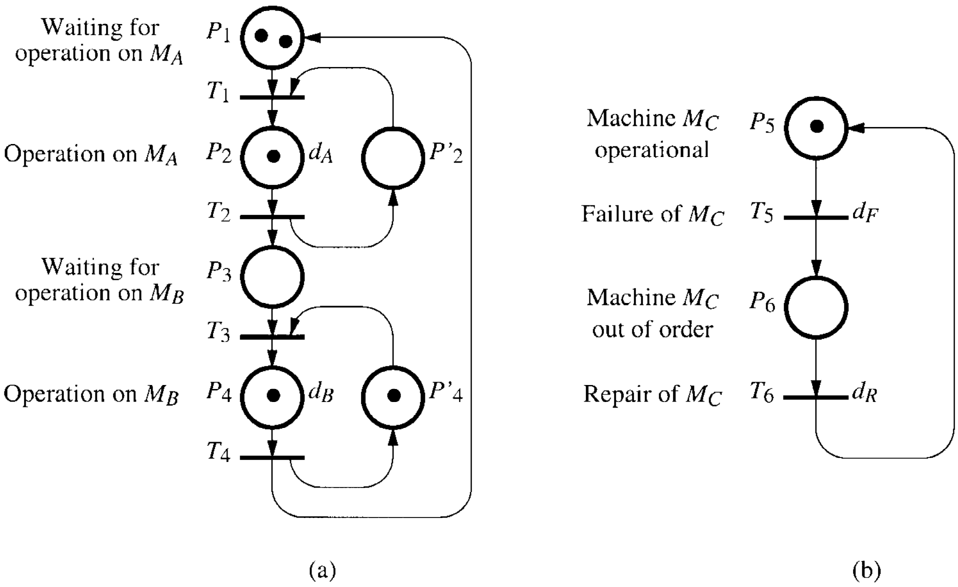

Figure 5. Natural modeling of time. (a) Modeling of timed operations. (b) Modeling of waited events.

transition T_2 (during this time, the token is said to be *unavailable*). Similarly, a token put into P_4 remains unavailable for d_B. It may be *convenient not to specify the delay associated with a place if this delay is zero*; in our example, any token put into P_1, P'_2, P_3, or P'_4, is immediately available. In Fig. 5.a, the times d_A and d_B may be either deterministic (P-timed PN) or stochastic (exponential distribution or any other distribution).

Figure 5.b represents the state of a machine M_C which can fail and be repaired. The token in P_1 means that the machine is operational: the transition T_1 can be fired and d_F represents the time when the failure will occur. Similarly, d_R represents the repair time. In this example, the times d_F and d_R are stochastic, with any distribution; if they are exponentially distributed, the rates (to failure and to repair) may be represented instead of the times (this is usual in stochastic PNs).

In Fig. 5.a, the durations are naturally associated with the places modeling the corresponding operations. In Fig. 5.b, the durations are naturally associated with the transitions fired when the corresponding events occur. Now, if we want to associate all the delays with the transitions, how can we modify the Fig. 5.a to meet this requirement? Two solutions are presented in the sequel.

The first solution consists of preserving the structure of the PN and of associating the delays to the transitions corresponding to the events "End of operation on M_A" and "End of operation on M_B." This solution is illustrated in Fig. 6.a (*all the transitions without explicit delay are immediate*, i.e., fired as soon as they are enabled).

The second solution consists of associating an operation with a transition. For our example, the sub-PN made of T_1, P_2, T_2, and the corresponding arcs, in Fig. 5.a, are replaced

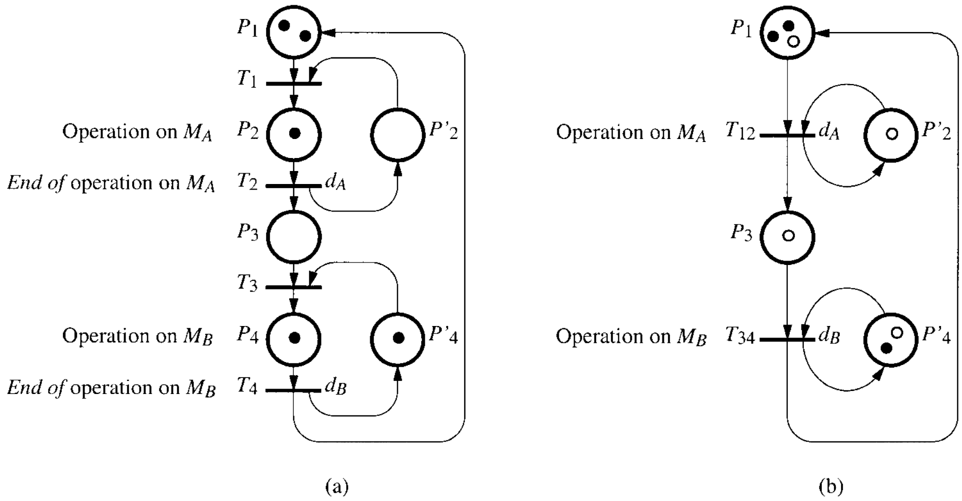

Figure 6. Time associated with the transitions. (a) Time to events. (b) Duration.

by the single transition T_{12} in Fig. 6.b. Similarly, transition T_{34} in Fig. 6.b represents the operation on M_B. This solution consisting of representing an operation by a transition in a discrete PN, although it is intuitively less natural, is often used because the obtained model contains less places and transitions. However, a problem remains: the representation of the tokens which were in the places that disappeared in the transformation from Fig. 5.a to Fig. 6.b. The token in P_2 on Fig. 5.a should be "in the transition T_{12}" in Fig. 6.b. In this figure, this is represented by *reserved* tokens (represented as white tokens) in the input places which have allowed the firing of T_1 in Fig. 5.a. In the model in Fig. 6.b, when T_{12} is enabled (at least one non-reserved token in both P_1 and P_2'), a token is reserved in both P_1 and P_2' (this reservation corresponds to the firing of T_1 in Fig. 5.a). The tokens are reserved for d_A, then T_{12} is fired (this firing corresponds to the firing of T_2 in Fig. 5.a): the reserved tokens are taken out and non-reserved tokens are put into P_3 and P_2'. Informally, the reserved tokens correspond to a token which "should be in the transition," hence they are not available for re-enabling a transition.

3.2. Modeling of Time in Hybrid Petri Nets

The authors have defined models of timed continuous and hybrid PNs in which time is associated with the transitions (however, other models could be used, based on the autonomous model defined in Section 2). *For the discrete part, the model illustrated in Fig. 6.a is used,*[6] i.e., times to events are associated with the corresponding transitions. *For the continuous part, the model draws inspiration from Fig. 6.b* and is illustrated in Fig. 7.

In Fig. 7.a, the operation on a machine is represented as a continuous flow on this machine. Since there is no discrete customer on the machine, there is no token "in the transition,"

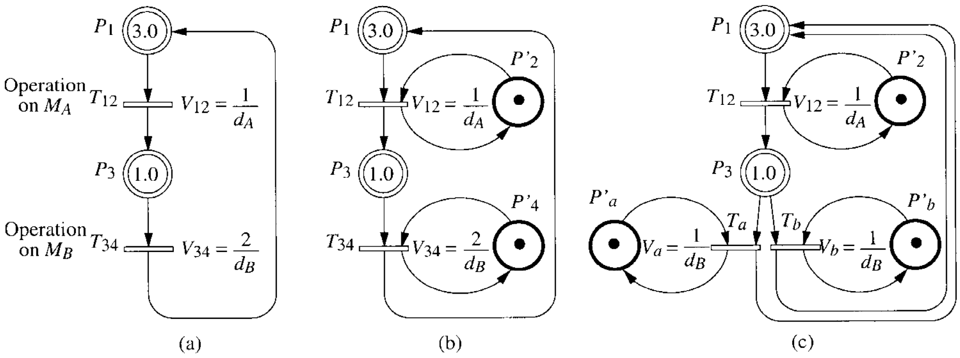

Figure 7. (a) Continuous PN approximating the behavior of the discrete PN in Fig. 6.b. (b) and (c) Equivalent hybrid PNs.

hence no need of a reserved mark (more precisely, we could say that there is an infinitely small part of token "in the transition"). If d_A is constant, the maximal firing rate of transition T_{12} in Fig. 6.b is $1/d_A$ (this maximal rate is obtained if there is always at least one token in P_1). Accordingly, the maximal speed associated with T_{12} in Fig. 7.a is $V_{12} = 1/d_A$. In Fig. 6.b, the maximal firing rate of transition T_{34} is $2/d_B$ since there are two servers (this maximal rate is obtained if there are always at least two tokens in P_3). Accordingly, the maximun speed associated with T_{34} in Fig. 7.a is $V_{34} = 2/d_B$. In Fig. 6.b, the maximal number of customers on a machine is fixed by places P_2' and P_4'. In Fig. 7.a, this limitation is taken into account by setting the maximal speeds V_{12} and V_{34}.

Figures 7.b and c present two hybrid PNs whose behaviors are similar to the behavior of the continuous PN in Fig. 7.a. It is clear that places P_2' and P_4' in Fig. 7.b add no constraint on the validation of the transitions. In Fig. 7.c, transition T_{34} is split into two transitions corresponding to the two servers of the machine. The behavior of this hybrid PN is such that $v_a + v_b = v_{34}$. For the system which is modeled here, it is clear that it is simpler to use the model in Fig. 7.a. Models similar to Fig. 7.b and c may be useful when a resource is shared between two (or more) productions. An example will be presented in Section 3.4 devoted to conflicts in hybrid PNs.

Up to now, the time associated with a D-transition and the maximal speed associated with a C-transition have not been specified. In Section 3.3, a model in which these times and speeds are constant will be presented. Other models where the speeds of the continuous part depend on time or other parameters, or where the times in the discrete part are stochastic, will be presented in Section 3.5.

3.3. Constant Timings and Speeds

In this section, the main ideas and concepts will be introduced progressively. First, a pure continuous system will be modeled by a continuous PN. Then, a discrete part will be added

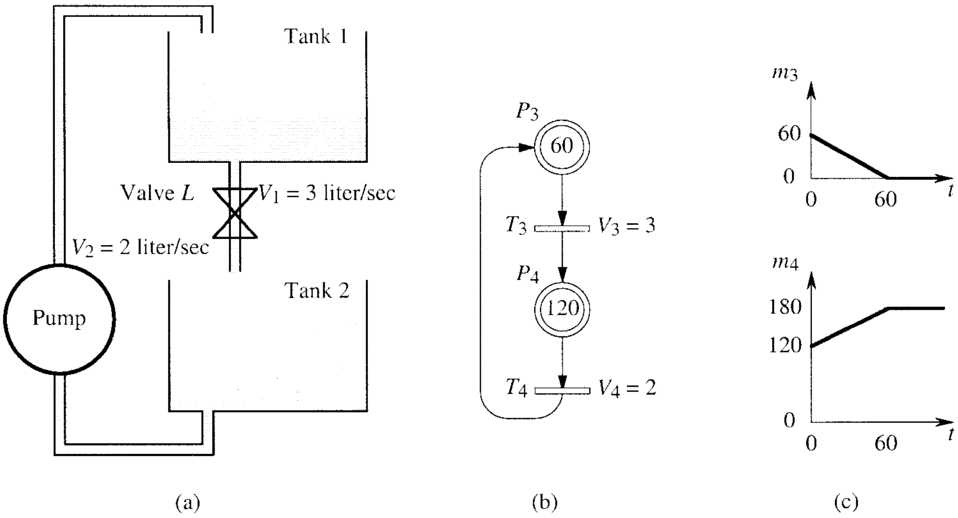

Figure 8. (a) A continuous system. (b) Continuous PN. (c) Marking evolution.

and the whole system will be modeled by a hybrid PN. The notions of IB-state (standing for "invariant behavior state" and evolution graph will be illustrated. Finally, some general topics relevant to a hybrid PN will be presented more formally.

3.3.1. *Example of Continuous Petri Net*

Figure 8.a represents a continuous system. A liquid flows from tank 1 to tank 2 by gravity (3 liter/sec). It is transferred from tank 2 to tank 1 by a pump (2 liter/sec). We assume that the liquid in the pipes is not taken into account and that, at initial state, there are 60 liters in tank 1 and 120 liters in tank 2. The behavior of this system is modeled by the timed continuous PN in Fig. 8.b where the markings in P_3 and P_4 represent the quantity of liquid in tanks 1 and 2, respectively; the speeds V_3 and V_4 associated with the transitions T_3 and T_4 correspond to the flows previously presented.[7]

Consider the model in Fig. 8.b. When $m_3 > 0$, transition T_3 is fired at the speed $V_3 =$ 3 liter/sec. When $m_4 > 0$, transition T_4 is fired at the speed $V_4 = 2$ liter/sec. It follows that

$$m_3(t + \mathrm{d}t) = m_3(t) + (2 - 3)\mathrm{d}t, \tag{2}$$
$$m_4(t + \mathrm{d}t) = m_4(t) + (3 - 2)\mathrm{d}t. \tag{3}$$

Since $m_3(0) = 60$ and $m_4(0) = 120$, from (2) and (3):

$$m_3(t) = 60 - 1\,t, \tag{4}$$
$$m_4(t) = 120 + 1\,t. \tag{5}$$

Equations (4) and (5) remain true as long as $m_3 > 0$.

At time $t = 60$, $m_3 = 0$ and $m_4 = 180$. Transition T_4 can still be fired at its maximal speed since $m_4 > 0$, but T_3 cannot. As a matter of fact $m_3 = 0$ (tank 1 is empty). However, m_3 is fed at speed V_4 by firing of T_4 (tank 1 is fed by the pumping). Then, transition T_3 can be fired at speed 2 (speed of the flow from tank 1 to tank 2), which is no longer the maximal speed.

Maximal speed is noted with a capital V, and instantaneous speed with a small v. Then,

$$\left.\begin{aligned} v_1(t) = V_1 = 3 \\ v_2(t) = V_2 = 2 \end{aligned}\right\} \text{for } 0 \ \le \ t < 60. \tag{6}$$

$$\left.\begin{aligned} v_1(t) = V_2 = 2 \\ v_2(t) = V_2 = 2 \end{aligned}\right\} \text{for } t \ \ge \ 60. \tag{7}$$

The corresponding markings are illustrated in Fig. 8.c. For $t \ge 60$, transition T_4 is *strongly enabled* (all its input places are non-empty), and transition T_3 is *weakly enabled* (all its input places which are empty are fed by firing of other transitions). The definition of enabling in a continuous PN was given in (David and Alla, 1987). This definition is easily extended to hybrid PNs (Def. 4 below).

3.3.2. Example of Hybrid Petri Net

Let us now modify the specified behavior of the process in Fig. 8.a. The valve L may be either open (flow 3 liter/sec) or closed (no flow). The pump may be working (flow 2 liter/sec) or not. The following behavior is assumed (with the same initial state). In turn, the valve is open and the pump is working. At the initial state, the valve is open for 90 seconds, then the pump is working for 75 seconds, the valve is open again for 90 seconds, and so on.

This system is modeled by the hybrid PN in Fig. 9. For the marking in this figure (initial time), transition T_3 is strongly enabled because P_3 is not empty and there is a token in P_1; hence it is fired at its maximal speed $v_3 = V_3 = 3$. At the same time, transition T_4 is not enabled because there is no token in P_2: $v_2 = 0$. It follows that P_3 is emptied at the speed of 3 liters per second: it becomes empty at $t = 20$ seconds. Hence, from $t = 20$, neither T_3 (because P_3 is empty) nor T_4 (because there is no token in P_2) is enabled. The only enabled transition is T_1 which will be fired at $t = 90$. Hence, from $t = 90$, both T_2 and T_4 become enabled: T_4 is continuously fired at $v_4 = V_4 = 2$ up to $t = 165$ when T_2 is fired. And so on.

3.3.3. IB-States and evolution graph

The marking of the hybrid PN in Fig. 9 is the vector $\mathbf{m} = (m_1, m_2, m_3, m_4)$. Because of the marking invariants $m_1 + m_2 = 1$ and $m_3 + m_4 = 180$, the marking is completely known from (m_1, m_3). This is illustrated in Fig. 10.a where A represents the initial marking. The evolution of the hybrid PN may be analyzed thanks to the evolution graph in Fig. 10.c. This graph is made of IB-states and transitions among them. An IB-state is such that *the marking of the discrete part and the instantaneous speed vector of the continuous part*

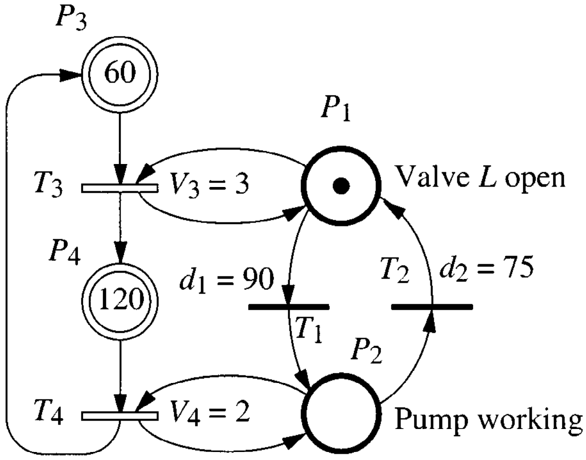

Figure 9. Timed hybrid Petri net.

remain constant as long as the system is in the same IB-state. For example, IB-state 1 in Fig. 10.c corresponds to the behavior from the initial state in Fig. 9: the marking of the discrete part is $(m_1, m_2) = (1, 0)$ and the instantaneous speed vector is $(v_3, v_4) = (3, 0)$. The marking of the continuous part which is written $(m_3, m_4) = (60, 120)$, corresponds to the continuous marking when the IB-state is reached (initial state in our example). The continuous marking evolves continuously and linearly as long as the system is in the same IB-state. For each C-place, the *balance* of the marking is defined as the algebraic sum of instantaneous speeds of the transitions *feeding* the place (i.e., input transitions, with a positive sign), and of the transitions *emptying* the place (i.e., output transitions, with a negative sign). Hence, this balance, denoted B_i for the place P_i, corresponds to the time derivative of m_i. For the C-places in Fig. 9:

$$\dot{m}_3 = B_3 = v_4 - v_3 \qquad \text{and} \qquad \dot{m}_4 = B_4 = v_3 - v_4. \tag{8}$$

For the IB-state 1, one obtains[8]

$$\dot{m}_3 = B_3 = 0 - 3 = -3 \qquad \text{and} \qquad \dot{m}_4 = B_4 = 3 - 0 = +3. \tag{9}$$

Accordingly, the marking m_3 decreases and becomes 0 at time $t = 20$ sec (trajectory from A to B in Fig. 10.a). The occurrence of this event "$m_3 = 0$" modifies the transition enablings in the hybrid PN. Hence, a new IB-state is reached. In Fig. 10.c, the transition between IB-state 1 and IB-state 2 is labelled by two pieces of information separated by a /. The information on the lefthand side is the *event* provoking the transition, and the information on the righthand side is the *time elapsed* in IB-state 1. When IB-state 2 is reached, the marking is $\mathbf{m} = (1, 0, 0, 180)$; the only transition enabled is T_1. The firing of this transition will occur at $t = 90$, i.e., after 70 seconds elapsed in IB-state 2. This firing corresponds to the jump from B to C in Fig. 10.a, and to the transition from IB-state

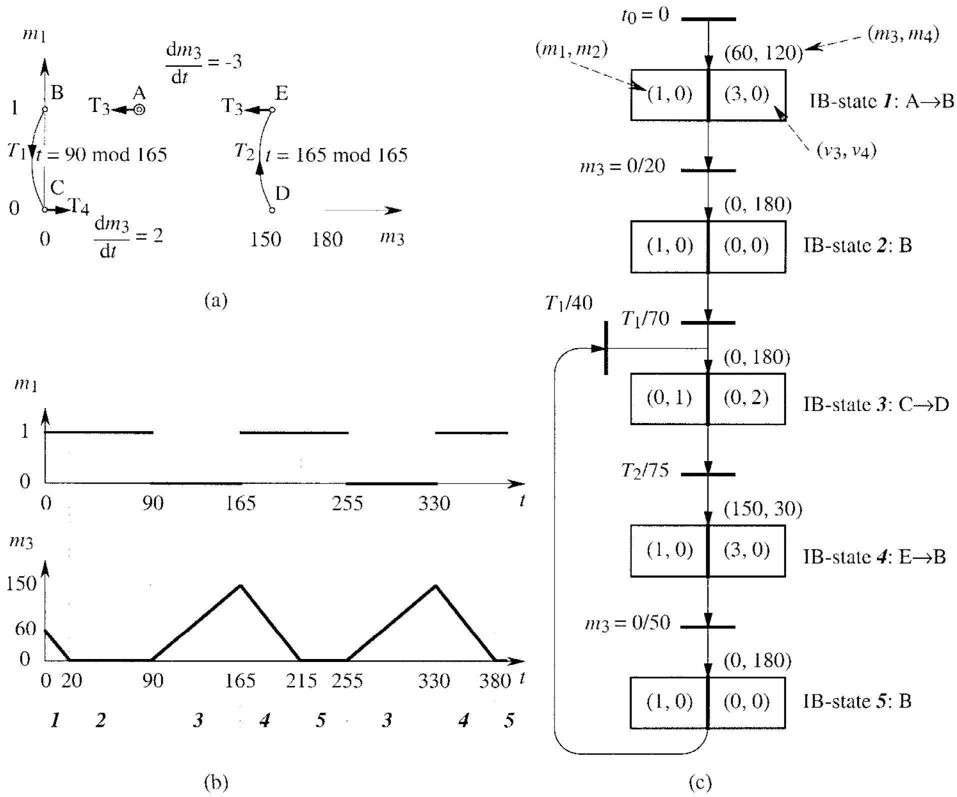

Figure 10. Evolution of the state of the hybrid PN in Fig. 9. (a) In the space of markings. (b) Markings. (c) Evolution graph.

2 to IB-state 3 in Fig. 10.c. And so on. Figure 10.c shows that, after a transient behavior (IB-state 1 and 2), a periodic behavior is reached (IB-state 3, 4 and 5). Figure 10.b presents the evolutions of markings m_1 and m_3 from $tg = 0$ up to t about 400.

3.3.4. *Formalization*

This section formalizes some concepts which were presented intuitively in the previous sections. Most of these results were given in (Le Bail *et al.*, 1991).

Definition 3 A **discrete transition** in a hybrid PN is **enabled** if every place P_i in ${}^{\circ}T_j$ meets the condition

$$m_i \geq Pre(P_i, T_j).$$

Definition[9] 4 A **continuous transition** T_j in a hybrid PN is **enabled** if every place P_i in ${}^\circ T_j$ meets the following conditions.

If P_i is a *discrete place*:

$$m_i \geq \text{Pre}(P_i, T_j).$$

If P_i is a *continuous place*:

either 1) $m_i > 0$,

or 2) P_i *is fed*, i.e., there is at least one continuous transition T_k in ${}^\circ P_i$ such that $v_k > 0$.

An enabled C-transition is **strongly enabled** if $m_i > 0$ for every continuous place in ${}^\circ T_j$.; it is **weakly enabled** otherwise. ■

Let us now give some helpful notations. The sets of places P^D and P^C, and the sets of transitions T^D and T^C, were defined in Def. 1 in Section 2.2.1. Let $\mathbf{m}^D$ and $\mathbf{m}^C$ denote the markings of the discrete part (places in P^D) and of the continuous part (places in P^C), respectively. It is always possible to order the places in the hybrid PN in such a way that the index i of any D-place P_i is always lower than the index k of any C-place P_k; in that way, $\mathbf{m} = (\mathbf{m}^D, \mathbf{m}^C)$. If the transitions are ordered similarly, the vector of instantaneous speeds may be denoted by $\mathbf{v} = (0, \mathbf{v}^C)$ (this vector whose dimension is $|T|$, the cardinality of T, contains $|T^D|$ 0's for the D-transitions; the dimension of $\mathbf{v}^C$ is $|T^C|$).

Definition 5 An **IB-state** corresponds to a time interval such that:

1) $\mathbf{m}^D$ is constant;
2) $\mathbf{v}^C$ is constant;
3) the set of discrete transition enablings is constant,[10]
4) when the IB-state is reached, $\mathbf{m}^C$ always has the same value.[11]

Definition 6 The **balance** of the marking of a continuous place P_i is:

$$B_i = \sum_{T_j \in {}^\circ P_i} Post(P_i, T_j) \cdot v_j - \sum_{T_k \in P_i^\circ} Pre(P_i, T_k) \cdot v_k \qquad (= \dot{m}_i). \tag{10}$$

Property 1 A change of IB-state can occur only if an **event** belonging to one of the following kinds occur.

First kind: a *discrete transition* is fired.

Second kind: the marking of a *continuous place* (whose balance is negative) becomes 0.

Third kind: the marking of a *continuous place* (whose balance is positive), that is an input place of a discrete transition, reaches the weight of the arc linking the place to the transition[12] (or a multiple of this weight in case of multiple enabling). ■

In Fig. 10.c, the transition between IB-state 1 and IB-state 2 is of the second kind and the transition between IB-state 2 and IB-state 3 is of the first kind.

Let n_j denote the number of firings of the D-transition T_j from the initial time. The vector of discrete firings may be denoted by $\mathbf{n} = (\mathbf{n}D, \mathbf{0})$; this vector is made of the $|T^D|$-dimensional vector $\mathbf{n}^D$ corresponding to the numbers of firings of every D-transition, and $|T^C|$ zeros corresponding to the C-transitions.

Let us now present the *fundamental equation when time is involved*. In the sequel, the time will be explicitly noted, for example $\mathbf{m}(t)$ denotes the marking $\mathbf{m}$ at time t.

Equation (1′) in Section 2.2.1 is modified in the following way. The characteristic vector $\mathbf{s}$ in Equ. (1′) may be specified as:

$$\mathbf{s}(t) = \mathbf{n}(t) + \int_0^t \mathbf{v}(u) \cdot \mathrm{d}u, \tag{11}$$

where the first term of the sum corresponds to D-transitions and the second term to C-transitions. Hence, Equ. (1′) becomes

$$\mathbf{m}(t) = \mathbf{m}(0) + \mathbf{W} \cdot \left(\mathbf{n}(t) + \int_0^t \mathbf{v}(u) \cdot \mathrm{d}u\right). \tag{12}$$

Let us consider for example the evolution of the hybrid PN in Fig. 9 from $t = 0$ to $t = 170$. During this time interval, T_1 is fired at $t = 90$, T_2 is fired at $t = 165$, T_3 is continuously fired at $v_3 = 3$ from $t = 0$ to $t = 20$ then from $t = 165$ to $t = 170$ (i.e., during 25 sec), and T_4 is continuously fired at $v_4 = 2$ from $t = 90$ to $t = 165$ (i.e., during 75 sec). Hence we have:

$$\begin{gathered}\mathbf{m}(0) = \begin{bmatrix} 1 \\ 0 \\ 60 \\ 120 \end{bmatrix}; \quad \mathbf{W} = \begin{bmatrix} -1 & +1 & 0 & 0 \\ +1 & -1 & 0 & 0 \\ 0 & 0 & -1 & +1 \\ 0 & 0 & +1 & -1 \end{bmatrix}; \\ \mathbf{n}(170) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \quad \int_0^{170} \mathbf{v}(t) \cdot \mathrm{d}t = \begin{bmatrix} 0 \\ 0 \\ 75 \\ 150 \end{bmatrix}.\end{gathered} \tag{13}$$

From (12) and (13), one obtains:

$$\mathbf{m}(170) = \begin{bmatrix} 1 \\ 0 \\ 135 \\ 45 \end{bmatrix}. \tag{14}$$

In a general case, $\mathbf{W} = \begin{bmatrix} \mathbf{W}^D & \mathbf{0} \\ \mathbf{W}^{CD} & \mathbf{W}^C \end{bmatrix}$, where $\mathbf{W}^D$ corresponds to arcs among discrete nodes, $\mathbf{W}^C$ corresponds to arcs among continuous nodes, and $\mathbf{W}^{CD}$ corresponds to arcs among C-places and D-transitions. Arcs among D-places and C-transitions corresponds to the submatrix $\mathbf{0}$, according to the restriction about these arcs in Def. 1.

When $\mathbf{W}^{CD} = \mathbf{0}$, the net is called an **elementary hybrid PN**. Figure 11 corresponds to an elementary hybrid PN, according to its incidence matrix in (13). In such a net, there is a decoupling between the discrete and the continuous parts (one part may influence the behavior of the other one, but there is no “transformation” of discrete marking into continuous marking or vice-versa). The P-invariants (T-invariants) of an elementary hybrid

PN are the linear combination of the P-invariants (T-invariants) of the discrete part, and of the P-invariants (T-invariants) of the continuous part.

Let us end this section with a property related to a periodical behavior.

Property 2 The existence of a T-invariant is a necessary condition for a periodical functioning of a hybrid PN. ■

As a matter of fact, if a periodical behavior of period d has been reached, from (12) one obtains $\mathbf{m}(t + d) = \mathbf{m}(t) + \mathbf{W} \cdot \mathbf{s}(d) = \mathbf{m}(t)$, where $\mathbf{s}(d)$ is the characteristic vector associated with the trajectory during a period; since $\mathbf{W} \cdot \mathbf{s}(d) = \mathbf{0}$, $\mathbf{s}(d)$ is a T-invariant.

For the example in Fig. 9, $(1, 1, 0, 0)$ and $(0, 0, 1, 1)$ are T-invariants corresponding respectively to the discrete and the continuous part. The periodic behavior appearing in Fig. 10.b, corresponds to the linear combination $\mathbf{s}(165) = (1, 1, 150, 150)$.

3.4. Conflicts in a Hybrid Petri Net

If there are conflicts, several behaviors are possible. The conflict resolution is interesting when time is involved in the model. As a matter of fact, an autonomous PN (discrete, continuous, or hybrid) models the set of all the possible behaviors.

Firing speeds (and thus evolution graphs) can be calculated only if some hypotheses are made on solutions to the conflicts (if conflicts exist, of course). We will consider here the conflicts specific to hybrid PNs: conflict between a continuous and a discrete transition, and between two continuous transitions.

3.4.1. Conflict between a Continuous and a Discrete Transition

Rule 1. If there is a conflict between a discrete transition and a continuous transition, *the discrete transition has priority over the continuous transition.*[13] ■

This rule is intuitively logical since the firing of a C-transition corresponds to a continuous working while the firing of a D-transition corresponds to a brutal change of state of the system. For example, in Fig. 9, at $t = 165$, both T_2 and T_4 can be fired since $m_4 = 30$, $m_2 = 1$, and the delay $d_2 = 75$ ends. It appears clearly from this example that transition T_2 must be fired even if T_4 is still enabled. The authors have never encountered an example for which Rule 1 would be a handicap for modeling.

3.4.2. Conflict Between Two Continuous Transitions

Two cases have to be considered: either the common place is continuous or it is discrete, as illustrated in Fig. 11.

In Fig. 11.a, place P_1 is empty. It is fed by the continuous firing of T_1 at $v_1 = 2$. According to Def. 4, both T_2 and T_3 are weakly enabled, but there is a conflict because $v_1 < V_2 + V_3$. Hence, all the solutions such that $v_2 + v_3 = v_1 = 2$ are possible. Note that there is no

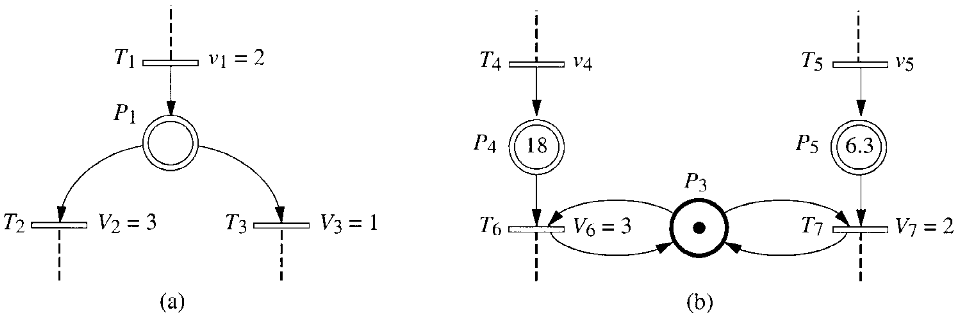

Figure 11. Conflicts between continuous transitions. (a) The common place is continuous. (b) The common place is discrete.

conflict if either $v_1 = 0$ (because T_2 and T_3 are not enabled) or $v_1 \geq V_2 + V_3$ (because T_2 and T_3 can be fired at their maximal speeds; if $v_1 > V_2 + V_3$, the balance is positive then P_1 does not remain empty).

Rule 2. For the models such that the maximal speeds do not depend on the markings of the input places:[14] if there is a *conflict between several continuous transitions with a common continuous input place* P_i which is empty, any solution such that the balance $B_i = 0$ is admissible. ■

Here are some examples. i) Priority to T_2 over T_3: $v_2 = 2$ and $v_3 = 0$. ii) Priority to T_3 over T_2: $v_3 = V_3 = 1$ and $v_2 = 1$. iii) Sharing proportional to maximal speeds: $v_2 = 1.5$ and $v_3 = 0.5$.

In Fig. 11.b, place P_3 corresponds to a resource which is shared between two operations represented by T_6 (operation L) and T_7 (operation R). If the resource is allocated to operation L, $v_6 = V_6 = 3$; in this case, the resource cannot make operation R, i.e., $v_7 = 0$. Similarly, if $v_7 = V_7 = 2$, then $v_6 = 0$. However, the resource may share the time between both operations abiding by

$$\frac{v_6}{V_6} + \frac{v_7}{V_7} = 1. \tag{15}$$

Note that there is no conflict if both P_4 and P_5 are empty and if they are fed at instantaneous speeds v_4 and v_5 such that $\frac{v_4}{V_6} + \frac{v_5}{V_7} \leq 1$.

Rule 3. For the models such that the maximal speeds do not depend on the markings of the input places: if there is a *conflict between several continuous transitions* $T_1, \ldots, T_a$, *with a common discrete input place* containing a token, any solution such that $\sum_{j=1}^{a} \frac{v_j}{V_j} = 1$ is admissible. ■

Here is an example of continuous system: a tap can mix hot water and cold water. The maximal flow is 0.2 liter/sec (when the tap is open and both waters are available). This

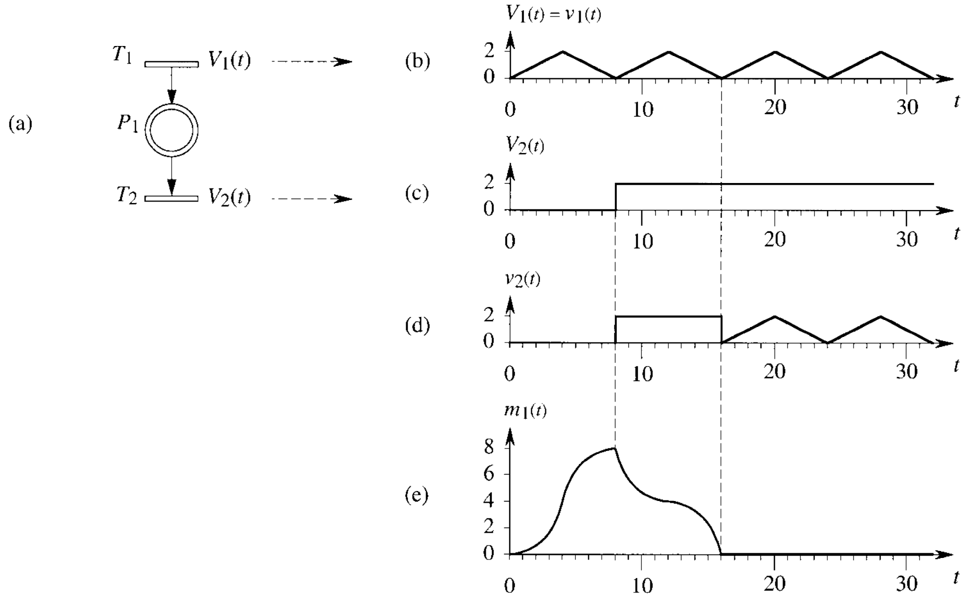

Figure 12. Continuous PN whose maximal firing speeds depend on time.

system is modeled like in Fig. 11.b, where P_3 represents the tap, T_6 and T_7 correspond to the flows of hot and cold water, respectively: $V_6 = V_7 = 0.2$. Here are some possible cases of behavior. i) Flow of hot water: $v_6 = 0.2$ and $v_7 = 0$. ii) Flow of cold water: $v_6 = 0$ and $v_7 = 0.2$. iii) Example of mixed temperature: $v_6 = 0.14$ and $v_7 = 0.06$ ($v_6 + v_7 = 0.2$ is verified, corresponding to Equ. (15) since $V_6 = V_7$ in this particular case).

3.5. Timing and Maximal Speed Depending on Time

Several examples are presented in the sequel. In the first one, maximal speeds are explicit functions of time. Then, examples where the maximal speeds depend on the marking are given. Finally, the last example illustrates a hybrid PN whose discrete part is stochastic. Some of the examples are continuous PNs, i.e., particular cases of hybrid PNs; adding a discrete part would complicate the presentation without necessity.

3.5.1. Maximal Speed Depending on Time

Figure 12.a represents a continuous PN whose maximal speeds depend on time. The maximal speed $V_1(t)$ associated with transition T_1 is periodic as illustrated in Fig. 12.b. The maximal speed $V_2(t)$ associated with transition T_2 is 0 for t up to 8 then $V_2(t) = 2$ for

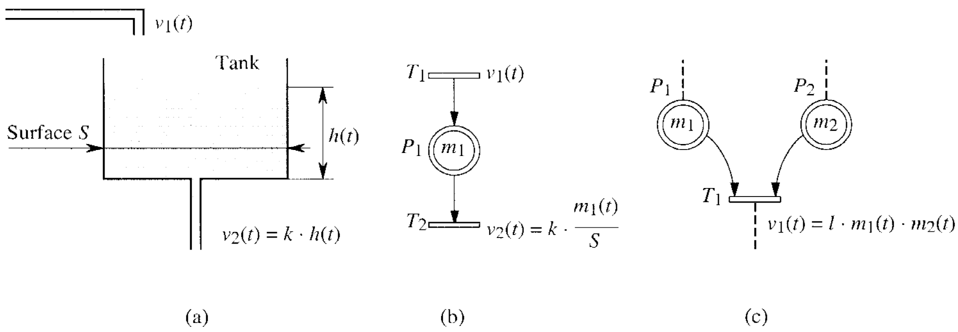

Figure 13. Other applications for the calculation of the firing speed of a transition.

$t \geq 8$ as illustrated in Fig. 12.c. The instantaneous firing speeds are no longer piecewise constant. It follows that the marking of a place is given by an integral.

According to Equ. (12) in Section 3.3.4, for the place P_1 in Fig. 12.a:

$$m_1(t) = m_1(0) + \int_0^t (v_1(u) - v_2(u))\mathrm{d}u \tag{16}$$

Since T_1 is always strongly enabled, $v_1(t) = V_1(t)$.

As long as $V_2(t) = 0$, $v_2(t) = 0$.

At $t = 8$, when $V_2(t)$ takes the value 2, $m_1(t) = 8$ (this is obtained from (16), given $m_1(0) = 0$). Thus T_2 is strongly enabled: $v_2(t) = V_2(t) = 2$ for some time.

The calculation shows that $m_2(t) = 0$ at $t = 16$. From this time, transition T_2 is weakly enabled because $V_1(t) \leq V_2(t)$; hence $v_2(t) = v_1(t) = V_1(t)$.

The values of $v_2(t)$ and $m_1(t)$ are illustrated in Figures 12.d and e.

Continuous Petri nets with maximal speeds depending on time are defined in (Dubois *et al.*, 1994), in which a simulation algorithm for the case of piecewise constant speeds is given.

3.5.2. *Maximal Speed Depending on Marking*

The different models presented above correspond to particular cases of the transition firing speed calculation. Nevertheless, the relation giving a marking in function of time, i.e., *Equ. (12), is true for any expression of* $\mathbf{v}(t)$ (however, the speeds and markings must remain non-negative). Other applications for the calculation of $\mathbf{v}(t)$ can be considered.

For example, the system presented in Fig. 13.a represents a tank filled with an input flow $v_1(t)$ and an output flow $v_2(t)$ which is proportional to the height of the liquid in the tank. This system can be modeled by the continuous PN of Fig. 13.b where the marking of place P_1 represents the volume of liquid in the tank and the firing speed of transition T_2 is $v_2(t) = k \cdot m_1(t)/S$ (S is the surface of the section of the tank).

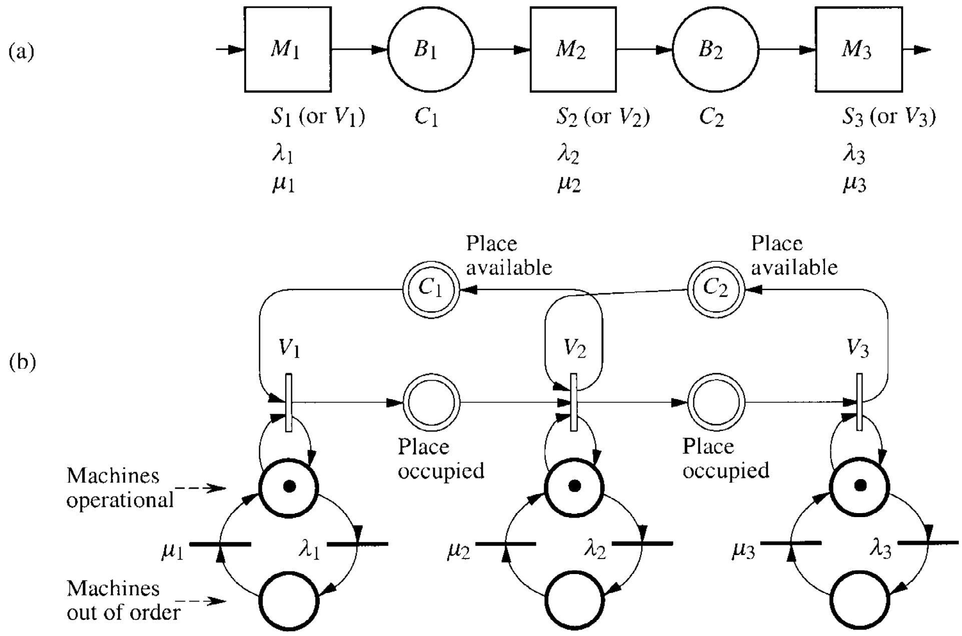

Figure 14. (a) Transfer line. (b) Hybrid PN with a stochastic discrete part, modeling the behavior of this system.

Another example of the calculation of the firing speed is given in Fig. 13.c. In this case, the firing speed depends on the product of the marking of the input places of the transition (Flaus,1996).

3.5.3. Stochastic Discrete Part

Figure 14.a shows the manufacturing line presented in the introduction. This is a very conventional model (Dallery *et al.*, 1989) (the number of machines and buffers may vary) in which the machines M_i are characterized by 3 parameters, a constant service time S_i, a failure rate λ_i and a repair rate μ_i; each buffer B_j is characterized by its capacity C_j. The number of states of such a system is considerable. A continuous model in which parts are assumed to move on the machines like a flow with a speed of $V_i = 1/S_i$, has been in use for a long time now (Zimmern, 1956). This continuous model is normally a very good approximation (David *et al.*, 1992).

When modeling a system we try to use general standard models such as Petri nets, queuing networks, automata, etc. Only a few years ago it was impossible to “include” the system described above in one of these models. This system has two features: 1) one part is discrete and the other is continuous; 2) one part is deterministic (processing of constant duration on a machine) and the other is stochastic (time between failures and between repairs). It can now be modeled like a hybrid PN with a stochastic discrete part (Dubois and Alla, 1993).

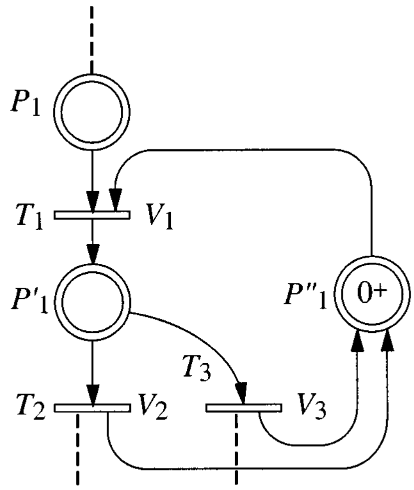

Figure 15. Use of the marking 0^+ for modeling the constraint $v_2 + v_3 = v_1$.

See Fig. 14.b. This modeling does not change the resolution method (Gershwin and Schick, 1980; Dubois and Forestier, 1982; Dallery *et al.*, 1989) but provides an interesting formal structure from a theoretical standpoint which could well lead to other ideas.

4. Application Examples

Several examples illustrating modeling by hybrid Petri nets have already been published. In (Alla *et al.*, 1992), a Motorola production system is modeled: a batch (discrete) is transformed and processed as a continuous flow. In (Alla, David, 1998), a water supply system is modeled: the continuous part of the model corresponds to storing and flow of water. In the sequel, two examples illustrate the use of the marking 0^+ and of the weight 0^+ introduced in Def. 2 (Section 2.3).

First Example

Figure 15 illustrates use of the symbol 0^+ as a marking of C-place. The modeled system, presented in (Balduzzi *et al.*, 1998), is as follows: the flow of parts processed by machine M_1 is routed to the machines M_2 and M_3, and this routing is immediate (one could also imagine a liquid flow separated into two flows). Place P_1 represents the input buffer of machine M_1, and transition T_1 represents the processing on this machine. Transitions T_2 and T_3 model the flows which are routed to machines M_2 and M_3, respectively. The loop containing places P'_1 and P''_1 expresses the constraint $v_2 + v_3 = v_1$ ($v_1 \leq V_1$, depending on the marking of P_1 and its feeding speed). If the sum of markings in P'_1 and P''_1 were

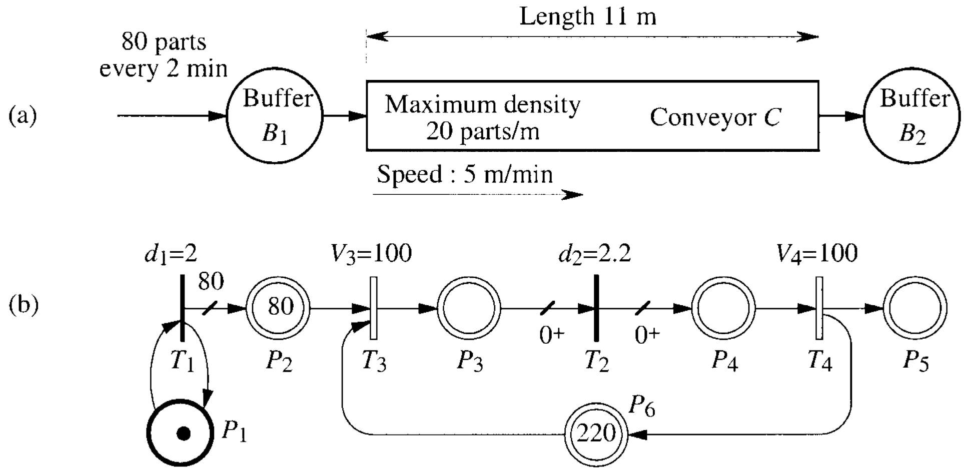

Figure 16. Modeling of a delay, using the weight 0^+. (a) System with a conveyor. (b) Extended hybrid PN modeling this system.

zero, none of the transitions among T_1, T_2, and T_3, would be enabled. Using conventional marking 0^+, an infinitely small quantity, allows enabling of these transitions and abiding to constraints on firing speeds.

Second Example

Figure 16.a represents[15] a conveyor C with an input buffer B_1 and an output buffer B_2 (both capacities of B_1 and B_2 are assumed not to be bounded). A batch of 80 parts is deposited into B_1 every 2 min from $t = 0$. The conveyor length is 11 m and its speed is 5 m/min. Its maximal capacity is 20 parts/m (i.e., 220 parts for the total length).

The behavior of this system is represented by the extended hybrid PN in Fig. 16.b where the flow of parts is considered as continuous. Markings of places P_2 and P_5 correspond to the parts in the input and output buffers. Transitions T_3 and T_4 correspond to parts entering and leaving the conveyor. The maximal speeds associated with these transitions correspond to:

$$V_3 = V_4 = \text{conveyor speed} \times \text{maximal density} = 5 \times 20 = 100 \text{ parts/min}. \qquad (17)$$

Place P_6 models the place available on the conveyor. The delay associated with transition T_2 corresponds to the time spent on the conveyor, i.e.,

$$d_2 = \frac{\text{Conveyor length}}{\text{Conveyor speed}} = \frac{11}{5} = 2.2 \text{ min}. \qquad (18)$$

The weights 0^+ of arcs from P_3 to T_2 and from T_2 to P_4 mean that: as soon as an infinitely small quantity of parts is put on the conveyor (i.e., in P_3), transition T_2 is enabled and this quantity will be put at the end of the conveyor (i.e., in P_4) when the time d_2 has elapsed. Roughly speaking, this model corresponds to a "*continuous firing of a discrete transition*."

Note that, if the conveyor speed changes, the values of V_3, V_4, and d_2, change immediately according to (17) and (18) (David and Caramihai, 2000).

5. Hybrid Petri Nets and Hybrid Automata

It has been shown in the previous sections that a hybrid PN inherits all the advantages of PN models. It provides models designed in an intuitive way. Meanwhile, except for the properties related to invariants, a quantitative analysis can be performed only via the construction of the evolution graph, which is a kind of simulation.

Hybrid automata are another tool for the modeling of hybrid dynamic systems. Several procedures have been proposed to analyze systems modeled by hybrid automata (Alur *et al.*, 1995). Although this analysis is oriented mostly towards the verification of system specifications, these procedures may also be used for other analysis goals. Hybrid automata are difficult to use for the modeling of complex systems. The goal of this section is to associate the modeling power of hybrid PNs with the analysis power of hybrid automata by an automatic transformation from a hybrid PN into a hybrid automaton. This approach is similar to the approach combining stochastic PNs with Markov chains (automatic transformation from a stochastic PN into a Markov chain for which powerful analysis methods exist).

5.1. Hybrid Automata

In order to introduce the hybrid automaton model intuitively, let us consider an example. The water level in a tank is controled through a monitor, which continuously senses the water level and turns a pump *on* or *off*. The water level is represented by a variable h. When the pump is *off*, the water level falls by 2 dm/min (decimeter per minute). When the pump is *on*, the water level rises by 1 dm/min. Suppose that initially the water level is 6 dm and the pump is turned *on*. We wish to keep the level of water between 1 and 12 dm. Moreover, there is a delay of 2 min from the moment that monitor signals the status of the pump until the change becomes effective. An extended hybrid PN describing the water level monitor is shown in Fig. 17.a. Places P_1 and P_2 represent respectively the status on and off of the pump and place P_3 the level in the tank. Transition T_3 is continuously fired at speed $v_3 = V_3 = 1$. As soon as $m_3 = 10$, transition T_2 is enabled; it is fired 2 minutes later: T_3 is no longer enabled and T_4 becomes enabled. Transition T_1 will be enabled as soon as $m_3 < 5 \ldots$

It is easy to see that this model is natural since each node represents a physical entity. The corresponding hybrid automaton is given in Fig. 17.b. The construction of this model is less intuitive. It has four locations: in locations l_1 and l_2, the pump is turned *on*, in locations l_3 and l_4, the pump is *off*. The variable y models time delays, $\dot{y}$ is the time derivative of y. Passing from l_1 to l_2 corresponds to enabling of T_2 (condition $h = 10$); passing from l_2 to l_3 corresponds to firing of T_2 (condition $y = 2$, and y is reset); and so on.

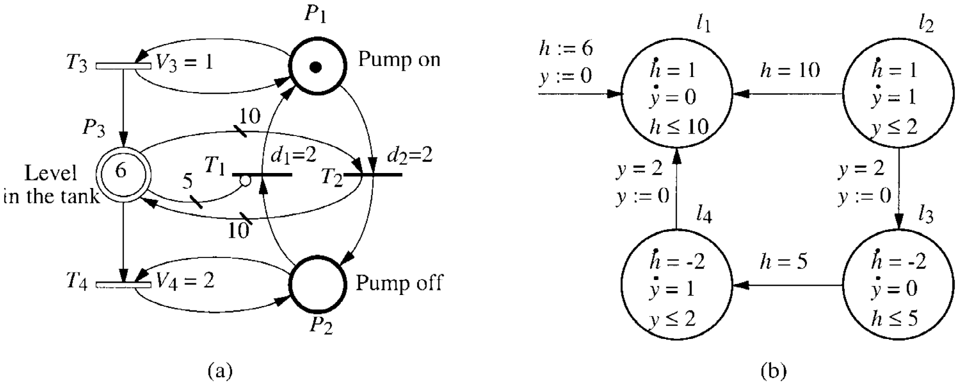

Figure 17. Models of the water level monitor: (a) Hybrid PN, (b) Hybrid automaton.

Adopting the terminology of (Alur *et al.*, 1995), a hybrid automaton is defined as follows (the examples refer to Fig. 17.b):

Definition 7 A **hybrid automaton** is a seven-tuple $H = \langle X, Q, \psi, Inv, A, Ev, As\rangle$:

X is a finite set of real-valued *variables* $\{x_i\}$. We denote by $\mathbf{x}$ the vector of variables. The values of all variables at a given moment define a *state of variables* (e.g.: $X = \{h, y\}$). The set of states of variables is denoted by V.

Q is a finite set of vertices called locations (e.g.: $Q = \{l_1, l_2, l_3, l_4\}$).

ψ is a function that assigns to each location $l \in Q$ a function ψ_i describing the evolution of variables as a function of time (e.g. in l_1: $\dot{h} = 1$ and $\dot{y} = 0\}$).

Inv is a function that assigns to each location $l \in Q$ a predicate Inv_l called the *invariant* of l (e.g. in l_1: $h \leq 10\}$).

A is a finite set of arcs called transitions. Each transition $a = (l, l')$ joins a source location $l \in Q$ to a target location $l' \in Q$ (e.g.: $A = \{(l_1, l_2), (l_2, l_3), (l_3, l_4), (l_4, l_1)\}$). An additional arc without source location corresponds to initialization.

Ev is a function that assigns to each transition $a = (l, l')$, a predicate Ev_a called *event*. The execution of transition $a = (l, l')$ is conditioned by the occurrence of Ev_a (e.g.: $Ev_{(l_2,l_3)}$ is "$y = 2$").

As is a function that assigns to each transition $a = (l, l')$ a relation As_a called *assignment*. It is used to model the discrete changing of the values of continuous variables. As_a: $x := g(x)$ (e.g.: $As_{(l_2,l_3)}$ corresponds to "$y := 0$"). ■

A particular class of hybrid automata is represented by **linear hybrid automata** which have the following features:

1) The evolution functions ψ_l within a location l are restricted to linear functions defined by a set of first-order differential equations of the form $\dot{\mathbf{x}} = \mathbf{k}_l$ with $\mathbf{k}_l$ denoting a constant vector.

2) Each location invariant Inv_l and transition event Ev_a are defined by a conjunction of linear inequalities over V of the form $\mathbf{Rx} + \mathbf{c} \leq \mathbf{0}$, where $\mathbf{R}$ is a constant matrix, $\mathbf{c}$ is a constant vector, and $\mathbf{x} \in V$.

3) The transition assignment As_a is defined by a relation $\mathbf{x} := V$, where $\mathbf{x} := V$, for α_x is a linear term over V of the form $\mathbf{R}$"$\mathbf{x} + \mathbf{c}'$, with $\mathbf{R}'$ is a constant matrix and $\mathbf{c}'$ is a constant vector.

5.2. From Hybrid PNs to Hybrid Automata

In order to systematize the change from hybrid PNs to hybrid automata, an algorithm has been developed for the construction of the hybrid automaton associated with a hybrid PN (Allam and Alla, 1998). Then quantitative analysis can be performed via the hybrid automaton model.

The hybrid PN functioning may be characterized by the *evolution vector*, made up of:

1) the *enabled* D-transitions (more precisely, all the validations since, at a time t, a transition may be enabled twice[16] or more)

2) the *balance* of the marking of the C-places (Def. 6 in Section 3.3.4).

According to Def. 5, *one can notice that an IB-state implies an evolution vector*. Hence, in the sequel, obtaining of the hybrid automaton is presented from the evolution graph. First, each IB-state is transformed into a location. Then, several locations corresponding to the same evolution vector may be merged.

In a hybrid PN (hence in an IB-state), the continuous variables are:

1) the residual time to firing for every enabling of a timed D-transition;

2) the marking of every continuous place.

Similarly, in every location, the continuous variables are:

1) the time elapsed since enabling for every enabling of timed transition (the "delay not yet elapsed" corresponds to the invariant of the location which is still satisfied, and "delay elapsed" is an event provoking a transition to another location;

2) the marking of every continuous place.

Let us illustrate this transformation from the hybrid PN in Fig. 17.a. The corresponding evolution graph is presented in Fig. 18.a. For construction of the hybrid automaton, the following continuous variables are considered.

1) A variable yj for every timed *D-transition* T_j. When T_j is enabled but not yet fired: $\dot{y} = 1$. (in case of multiple enabling, new variables will be added when necessary: $y_{j,1}, y_{j,2}, \ldots$).

2) A variable m_i for every *C-place* P_i; at any time $\dot{m} = B_i$ (Section 3.3.3).

For our example, the evolution vector is $(\dot{y}_1, \dot{y}_2, \dot{m}_3)$ and the hybrid automaton obtained from Fig. 18.a is shown in Fig. 18.b. At initial time, $(y_1, y_2, m_3) = (0, 0, 6)$.

With the initial IB-state 1, is associated the initial location l_1. Neither T_1 nor T_2 is enabled, hence $\dot{y}_1 - \dot{y}_2 = 0$. Since $v_3 = 1$ and $v_4 = 0$, $B_3 = \dot{m}_3 = 1$. According to Fig. 18.a, the next event provoking a change of IB-state is "m_3 reaches the value 10". This is the event associated with the transition from l_1 to l_2 in Fig. 18.b, and the invariant associated with l_1 is $m_3 \leq 10$.

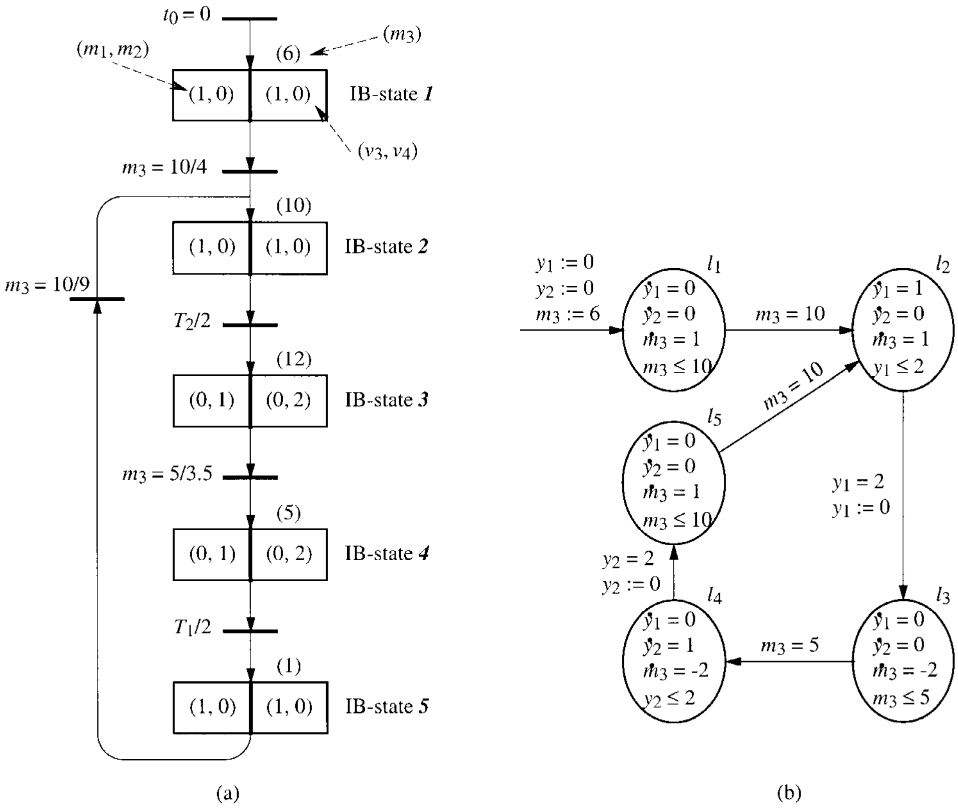

Figure 18. (a) Evolution graph for the hybrid PN in Fig. 17.a. (b) Corresponding hybrid automaton.

With the initial IB-state 2, is associated the initial location l_2. In this location $\dot{y}_1$ since T_1 is enabled and $\dot{m}_3$ is still 1. The next event will be $y_1 = 2$ (delay 2 associated with transition T_1 elapsed). Transition from l_2 to l_3 is performed and the value of y_1 is reset. And so on.

One can see in Fig. 18.b, that $(\dot{y}_1, \dot{y}_2, \dot{m}_3) = (0, 0, 1)$ in both l_1 and l_5. In addition, the invariant is the same and the event associated with the transitions to l_2 is the same. These locations may be merged. It can be noticed that, after the merging, this automaton is isomorphic to the automaton of Fig. 17.b. Only remaining difference: one clock y in Fig. 17.b and two clocks y_1 and y_2 in Fig. 18.b. (y_1 and y_2 are never activated simultaneously).

6. Conclusion

Continuous systems together with their modeling, analysis and control have long provided research with subject matters. Modeling, analysis and control of discrete systems have undergone major developments in recent decades. General models such as automata, Petri nets and queuing networks are used with variants (extensions).

In recent years the need has emerged to consider systems which are partially continuous and partially discrete, in other words hybrid systems. To model such systems we could use certain modeling tools normally used for discrete systems. This has in fact been done for automata (Alur *et al.*, 1995), and for Petri nets as shown above (Le Bail *et al.*, 1991). A software tool for modeling and simulation of continuous and hybrid PNs has been developed.[17]

Petri nets are known to be powerful tools for modeling and analysis of discrete systems. The continuous and hybrid Petri nets, extensions of the basic model, allow modeling and analysis of continuous and hybrid systems on the same conceptual basis. One can transform a timed hybrid PN into a hybrid automaton, then use the power analysis methods developed in the context of hybrid automata.

Notes

1. Since all the systems considered are *dynamic ones*, this adjective may be implicit. The expression *discrete systems* is used instead of 'discrete event systems' or 'discrete event dynamic systems', for homogeneity with the expressions 'continuous systems' and 'hybrid systems' (in *discrete event systems* the word event was added to avoid confusion with discrete-time systems, sometimes called 'discrete systems' by abuse of language).
2. The terms "mark" and "token" are normally synonymous. We use the word "mark" to refer either to continuous or to discrete marking. The word "token" always has a discrete meaning.
3. In the text, two components of a vector are separated by a comma. Curly brackets represent a column vector, i.e. $(a, b) = [a \quad b]^T = \begin{bmatrix} a \\ b \end{bmatrix}$.
4. $\mathcal{R}^+$ denotes the set of positive real numbers, including zero, and $\mathcal{N}$ denotes the set of natural numbers.
5. In (Alla and David, 1998), we have proposed for the case where the inhibitor arc has its origin at a continuous place: enabling if $m_i \leq r$ and zero test for $r = 0$. This convention is not satisfactory since an arc with a *zero weight* usually corresponds to the *absence* of arc. This is the reason why we introduce now the concept of infinitely small weigth. It also allows to have the same condition $m_i < r$ for either a D-place or a continuous place (and no longer $m_i \leq r$ for the continuous case).
6. In previous papers, the authors have used the model "with reserved tokens" (like Fig. 6.b) for the discrete part of a hybrid PN. From now on, the model "without reserved tokens" (like Fig. 6.a) will be used. The modeling power is the same for both models, i.e., any system modeled by one of these models can be modeled by the other one. Since, in a hybrid PN, the operations are essentially represented by continuous transitions, it may be convenient to model the discrete part by the model without reserved tokens. Note that both models (with or without reservation of tokens) behave similarly as long as there is no conflict.
7. Note that the unit for a quantity of liquid is arbitrary: for example, the unit is 1 liter in Fig. 8.b but it could be, for example, 1 m^3; in this case, we would have $m_3 = 0.060$ m^3 and $V_3 = 0.003$ m^3/sec (more generally, the behavior is similar if all the markings and all the speeds are multiplied by an arbitrary positive value). On the other hand, the unit is not arbitrary for a number of discrete objects (number of parts in a buffer for example).
8. Obviously, these notions apply to pure continuous PN as well. Equ. (1) and (2) in Section 3.3.1 correspond to $\dot{m}_3 = B_3 = 2 - 3$ and $\dot{m}_4 = B_4 = 3 - 2$.
9. This is an extension to hybrid PNs of the definition given in (David and Alla, 1987) for the continuous PNs.
10. This means that, if a D-transition T_j becomes enabled (or becomes enabled once more) because the marking of a C-place in $^\circ T_j$ has reached a sufficient value, a new IB-state is reached.
11. For example, in Fig. 10.c, IB-state ***3*** can be reached either from ***2*** or from ***5***. In both cases, $\mathbf{m}^C = (0, 180)$.
12. For a D-place, this case corresponds to a first kind event.

13. The choice represented by this rule was explicitly introduced in (David, 1997).
14. For some models, a transition firing speed may depend on the markings of the input places. For example, the variable speed continuous PN (David and Alla, 1990) or the models in Fig. 13.b and c.
15. This example is inspired from an example in (Demongodin and Prunet, 1992).
16. For example, a D-transition with a single input D-place containing two tokens is enabled twice. If the delay associated with the transition is not zero, the second enabling may occur between first enabling and first firing.
17. This software called SIRPHYCO is available at the address of one of the authors.

References

Alla, H., Cavaillé, J.-B., Le Bail, J. and Bel, G. 1992. Les systèmes de production par lot: une approche discret-continu utilisant les réseaux de Petri hybrides. *Symposium ADPM '92*. Paris.

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

Allam, M. and Alla, H., 1997. Modelling production systems by hybrid automata and hybrid Petri nets. *Conf. on Control of Industrial Systems* Belfort.

Allam, M. and Alla, H., 1998. From hybrid Petri nets to automata, *JESA* 32(9-10): 1165–1185.

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

Balduzzi, F., Giua, A. and Menga, G. 1998. Hybrid stochastic Petri nets: Firing dpeed computation and FMS modelling. *WODES '98* Gagliari.

Brinkman, P. L. and Blaauboer, W. A. 1990. Timed continuous Petri nets: A tool for analysis and simulation of discrete event systems. *European Simulation Symposium* Ghent.

Charbonnier, F. 1993. Etude et résolution des conflits dans les réseaux de Petri hybrides. Internal Report, LAG, INPG, Grenoble.

Ciardo, G., Nicol, D. and Trivedi, K. S. 1997. Discrete-event simulation of fluid stochastic Petri nets. *Petri Nets & Performance Models PNPM'97* Saint Malo, France, pp. 217–225.

Cohen, G., Gaubert, S. and Quadrat, J.-P. 1995. Asymptotic throughput of continuous timed Petri nets. *Conference on Decision and Control* New Orleans.

Cohen, G., Gaubert, S. and Quadrat, J.-P. 1998. *Algebraic System Analysis of Timed Petri Nets. Idempotency*. J. Gunawardena (Ed.). Cambridge Cambridge: University Press.

Dallery, Y., David, R. and Xie, X. 1989. Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Trans. on Computers* 34(9): 943–953.

David, R. 1997. Modeling of hybrid systems using continuous and hybrid Petri nets. *Petri Nets & Performance Models (PNPM'97)*. Saint Malo, France, pp. 47–58.

David, R. and Alla, H. 1987. Continuous Petri nets. *8th European Workshop on Application and Theory of Petri Nets* Zaragoza.

David, R. and Alla, H. 1990. Autonomous and timed continuous Petri nets. *11th International Conference on Application and Theory of Petri Nets* Paris, pp. 367–386.

David, R. and Alla, H. 1992. *Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems*. London: Prentice Hall Int.

David, R. and Caramihai, S. 2000. Modeling of delays on continuous flows thanks to extended hybrid Petri nets. *Int. Conf. an Automation of Mixed Proceses (ADPM 2000)*, Dortmund.

David, R., Xie, X. and Dallery, Y. 1990. Properties of continuous models of transfer lines with unreliable machines and finite buffers. *IMA Journal of Math. Applied in Business & Industry* 6: 281–308.

Demongodin, I. and Prunet, F. 1992. Extension of hybrid Petri nets to accumulation systems. *IMACS Int. Symp. on Mathematical Modelling and Scientific Computing* Bangalore, India.

Demongodin, I., Caradec, M. and Prunet, F. 1998. Fundamental concepts of analysis in batches Petri nets. *Int. IEEE Conf. on Systems, Man, and Cybernetics* San Diego, pp. 845–850.

Demongodin, I. and Koussoulas, N.T. 1998. Differential Petri nets: Representing continuous systems in a discrete event world. *IEEE Transactions on Automatic Control* 38(4).

Dubois, E. and Alla, H. 1993. Hybrid Petri nets with a stochastic discrete part. *European Control Conference* Groningen.

Dubois, E., Alla, H. and David, R. 1994. Continuous Petri net with maximal speeds depending on time. *4th Int. Conf. of RPI, Computer Integrated Manufacturing and Automation Technology* Troy, USA.

Dubois, D. and Forestier, J.-P. 1982. Productivité et en-cours moyens d'un ensemble de deux machines séparées par un stock. *RAIRO Automatique* 16(2): 105–132.

Flaus, J.-M. 1996. Hybrid flow nets for batch process modelling. *CESA 96, IEEE SMC*. Lille.

Gershwin, S. B. and Schick, I. C. 1980. Continuous Model of an Unreliable Two-Stage Material Flow System With a finite Interstage Buffer. *Technical report MIT LIDS-R- 1032*.

Halbwachs, N, Proy, Y. E. and Raymond, P. 1993. Verification of linear hybrid systems by means of convex approximations. *Proc. 5th Conf. on Decision and Compute-Aided Verification, LNCS 697* Springer, pp. 220–228.

Le Bail, J., Alla, H. and David, R. 1991. Hybrid Petri nets. *European Control Conference* Grenoble, pp. 1472–1477.

Le Bail, J., Alla, H. and David, R. 1993. Asymptotic continuous Petri nets. *Discrete Event Dynamic Systems: Theory and Applications* 2: 235–263.

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

Murata, T. 1989. Petri Nets: Properties, Analysis and applications. *Proceedings of the IEEE* 77(4): 541–580.

Olsder, G. J. 1993. Synchronized Continuous Flow Systems. In: *Discrete Event Systems: Modeling and Control*, S. Balemi, P. Kozak, & R. Smadinga (Eds.) Basel: Birkhäuser Verlag.

Peterson, J. L. 1981. *Petri Net Theory and the Modelling of Systems*. Prentice Hall.

Pettersson, S. and Lennartson, B. 1995. Hybrid modelling focused on hybrid Petri nets. *European Workshop on Hybrid Systems*. Grenoble, pp. 303–309.

Ramchandani, C. 1973. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. Ph.D., MIT, USA.

Sifakis, J. 1977. Use of Petri Nets for Performance Evaluation. In: *Measuring, Modelling and Evaluating Computer Systems*, H. Beilner and E. Gelenbe (Eds.), North-Holland, pp. 75–93.

Trivedi, K. S. and Kulkani, V. G. 1993. FSPNs: Fluid stochastic Petri nets. *14th International Conference on Application and Theory of Petri Nets*, Chicago.

Weiting, R. 1996. Hybrid high-level Nets. *Proceedings of the Winter Simulation Conference*, Coronado, USA, pp. 848–855.

Weiting, R. 1996. Modeling and Simulation of Hybrid Systems Using Hybrid High-Level Nets. *Proceedings of the 8th European Simulation Symposium*, Genova, pp. 158–162.

Zimmern, B. 1956. Etude de la propagation des arrÍts aléatoires dans les chaÓnes de production. *Revue de Statistique Appliquée* 4: 85–104.