Continuous Petri nets and hybrid automata: two bisimilar models for the simulation of positive systems

Latéfa Ghomri*

MELT (Manufacturing Engineering Laboratory of Tlemcen),
PB 213 Faculty of technology;
University of Tlemcen,
Algeria
Email: ghomri@mail.univ-tlemcen.dz
*Corresponding author

Hassane Alla

GIPSA-LAB (Grenoble, Image, Parole, Signal, Automatique),
University of Grenoble,
11 rue des Mathématiques, Saint Martin d'Heres, France
Email: Hassane.alla@gipsa-lab.grenoble-inp.fr

Abstract: Petri nets (PNs) are a well-known modelling tool for discrete event systems. Continuous PN were introduced in order to avoid the combinatory explosion of the number of states, when considering real life systems. The constant speed continuous Petri net (CCPN) can be used to model discrete events systems; in that case, they constitute an approximation, which is often satisfactory. They can also model positive continuous systems. Hybrid automata (HA) are a less compact and expressive model, but, they can be used to perform powerful analysis. In this paper, we first present deeply the continuous PN and its modelling advantages. Then we present the main contribution of this paper, that is a structural translation algorithm from a CCPN into a HA. The translation algorithm is structural in the sense that it does not depend on the initial marking of the Petri net. We prove the timed bisimilarity between both models.

Keywords: discrete event systems modelling; constant speed continuous Petri nets; CCPN; hybrid automata; bisimulation.

Reference to this paper should be made as follows: Ghomri, L. and Alla, H. (2018) 'Continuous Petri nets and hybrid automata: two bisimilar models for the simulation of positive systems', *Int. J. Simulation and Process Modelling*, Vol. 13, No. 1, pp.24–34.

Biographical notes: Latéfa Ghomri is an Associate Professor in Industrial Engineering and Control Engineering at Tlemcen University. Her PhD thesis focuses on supervisory control of hybrid dynamical systems. Her fields of interest are: simulation and modelling in manufacturing systems; supervisory control and discrete events systems.

Hassane Alla is a Professor at Grenoble University and researcher in Gipsa-Lab. His research is mainly concerned with tools derived from Petri nets used for the performance evaluation and for the control synthesis of discrete event systems. He is author or co-author of 130 publications. One of its main publications is a book on continuous and hybrid Petri nets which has been published in English and in French.

This paper is a revised and expanded version of a paper entitled 'Continuous Petri nets and hybrid automata for the analysis of manufacturing systems' presented at 15th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2015, Canada, Ottawa, 11–13 May 2015.

1 Introduction

Modelling and control of physical systems are crucial issues. In this work, we are interested in a particular class of systems, such as transport systems, manufacturing systems, communication systems, biological systems…, etc. These systems have in common that they are positive dynamic systems, i.e., the state is always positive. This class of systems often uses for their description continuous time models, like differential equations. However, they can also be approximated as discrete event systems (DESs) models, as it will be shown in the sequel. Simulation and formal

methods can be used for the quantitative analysis of these systems. Although several authors (van Dijk, 1999) demonstrate the absence of opposition between the formal methods and simulation and that both approaches are complementary, it is, generally, harder to use formal tools than simulation in practice, since the support of formal tools is difficult to use. In the past, we could only find in the literature related to formal tools, a few case studies. However recently, formal methods have considerably evolved. Industry adopts techniques such as model checking (Cousot and Cousot, 2010), in addition to simulation. Currently, we can find in the literature more real life industrial systems (Wainer, 2016).

The basic tools for modelling and analysing DESs are finite state automata or Petri nets (PNs). Some authors use similar models such as the discrete event systems specification (DEVS) (Norazura et al., 2015) or hybrid simulation (Lohmann et al., 2009). In this work, we are concerned with PNs since they provide a smart modelling showing in a graphical way the synchronisation, resource sharing and parallelism concepts. PNs are widely used to model DESs. 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. However, 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 PNs. Numerous authors have used the fluidification concept in order to solve the state explosion problems observed in DESs. In Ciardo et al. (1997), fluid stochastic PNs are introduced to perform a discrete event simulation for some continuous systems. Formal aspects on PN fluidification are developed in Recalde and Silva (2001), and the basic PN properties are analysed in comparison with the fluidified behaviour.

Continuous PNs were introduced by Alla and David (1998). This model constitutes a solution for the state explosion problems observed in DESs models. It is also a good model for describing positive continuous systems. In a continuous PN, the marking takes real values and the transitions firing is a continuous process. Continuous PNs inherit all the advantages of PNs model (concurrency, resource sharing, and synchronisation). Unlike autonomous continuous PNs where time is not involved, timed continuous PNs are a class of models where time is considered in the form of firing speeds associated with transitions. The nature of transitions firing speeds distinguishes the different types of timed continuous PNs formalisms. We consider here constant speed continuous Petri nets (CCPN in continuation) where a maximal constant firing speed is associated with each transition. The dynamic behaviour of the model is piecewise linear and event driven. CCPN is the basic and the most studied model in the class of continuous PNs formalisms. It gives often a good approximation for the modelling of DESs such as production systems or transportation systems. In Balduzzi et al. (2000), a model derived from the CCPN was defined, it is a slightly different model used for the control of DESs. The main difference is that minimal and maximal speeds are associated with transitions, even if for real life systems, the minimal value is always equal to zero. In Demongodin and Koussoulas (1998, 2008), a continuous approximation of a timed discrete PN by discretising time associated with transitions is introduced. This new model constitutes a continuous approximation and can be used for simulation, however the event driven aspect is lost. In Brinkman and Blaauboer (1990), timings are associated with places in order to model pure delays in manufacturing systems. For example, the parts flow on a conveyor can be easily described by a delay corresponding to the time required to reach the extremity of the conveyor. In case of infinite server semantics and variable maximal speeds associated with transitions, a formal state representation is possible as in linear continuous system theory. Then analytical results have been established such as deadlock-freeness (Júlvez et al., 2006; Mahulea et al., 2009) and quantitative analysis (Kara et al., 2008).

Hybrid automata (HA in the sequel) were introduced in Alur et al. (1995) and Alur and Dill (1994). They are the most general formalism to model hybrid dynamic systems, in the sense that they can model the largest range of dynamics for this class of systems. The relative ease of computing the reachable state space of some sub-models of HA is the strengths of this tool. This analytical formulation allows to compute formally the performance evaluation of the modelled systems and to synthesise controllers.

The aim of this work is first to show the modelling tool of the CCPN and then to prove formally the translation of a CCPN into a HA, thus allowing to deduce equivalent properties. In other words, we aim to characterise the HA which is bisimilar to CCPN. We will firstly present some basic notions of the CCPN model such as the macro marking and the invariant behaviour state. Then, a structural algorithm allowing translation of a CCPN into a HA will be given. We mean by structural the fact that the translation does not depend on the initial marking of the CCPN. The soundness of the translation algorithm is proved by demonstrating the bisimilarity between the CCPN and its equivalent HA.

This paper is organised as follows: In Section 2, we will present the CCPN formalism with its main properties. We will characterise the class of systems that a CCPN can model. In Section 3, we will present an algorithm allowing the translation of a CCPN into a HA. We will give the semantics of the CCPN and the HA resulting from the translation algorithm. In Section 4, we will prove the soundness of the translation algorithm; and in Section 5, a water supply system will illustrate the results of the proposed approach. We present a discussion on the usefulness of our contribution in Section 6 and we place our work compared to existing results in the same field. Finally, in Section 7 we will conclude the paper.

2 Constant speed continuous Petri net

Continuous PNs is a generic name of a class of formalisms. These formalisms are a result of an extension of classical (discrete) PNs. They are obtained from the discrete model by fluidifying the marking. Advantages of continuous PNs are mainly related to computability and decidability (Recalde et al., 2007). In the basic book containing the last results on continuous PNs (David and Alla, 2010), they are classified according to transitions firing speeds. The most basic model is CCPN. In this model, with each transition is associated a maximal firing speed, which confers to the model a piecewise linear behaviour and the dynamic remains event driven.

In the sequel, we define CCPN and some notions related to its behaviour. We suppose that the basic knowledge on classical (discrete) PN is acquired. For a complete presentation, see David and Alla (2010).

Definition 1 (CCPN):

A CCPNPN is a 6-tuple PN = (**P**, **T**, Pre, Post, V, $\mathbf{m_0}$) such that:

- $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ is a finite set of places
- $\mathbf{T} = \{T_1, T_2, \ldots, T_m\}$ is a finite set of transitions
- Pre: $\mathbf{P} \times \mathbf{T} \rightarrow \wp^+$ and Post: $\mathbf{P} \times \mathbf{T} \rightarrow \wp^+$ are, respectively, the backward and forward incidence mappings
- V: $\mathbf{T} \rightarrow \Re^+$ associates a maximal firing speed V_j with each transition T_j
- $\mathbf{m_0}$: $\mathbf{P} \rightarrow \Re^+$ gives the real-valued initial marking.

□

where $\wp^+$ and $\Re^+$ are respectively the set of positive rational numbers and positive real numbers.

We will adopt in the sequel the following notation:

- $^\circ P_i$ is the set of input transitions of place P_i
- P_i° is the set of output transitions of place P_i
- $^\circ T_j$ is the set of input places of transition T_j
- T_j° is the set of output places of transition T_j.

Let us now give an example in order to illustrate the CCPN smart modelling.

Example: Consider the manufacturing system schematised in Figure 1. This system contains three machines M_1, M_2 and M_3, preceded, respectively, by buffers B_1, B_2 and B_3. Pallets support the parts to be manufactured (each pallet supports a single product) first on M_1, then M_2 and finally, on M_3. After that, the pallets return to buffer B_1. Machines M_1, M_2 and M_3 can maximally manufacture, respectively, four and two and one parts per time unit. Let us suppose that the number of pallets in the system is 12 and that all the pallets are initially in B_1. The buffers, capacities are supposed infinite. This manufacturing system is a DES since machines and parts are *indivisible objects*. We approximate the modelling of its behaviour by a CCPN. Figure 2 gives the CCPN model of the manufacturing system. To differentiate the representation of continuous and discrete PNs, we use double lines to draw continuous places and transitions, whereas single lines are used in the case of discrete places and transitions. Transitions T_1, T_2 and T_3 represent, respectively, the machines M_1, M_2 and M_3. The maximal firing speeds associated with these three transitions are respectively 4, 2 and 1. The marking of places P_1, P_2 and P_3 (i.e., M_1, M_2 and M_3) represent, respectively, the number of pallets in the buffers B_1, B_2 and B_3.

□

Figure 1 A manufacturing system

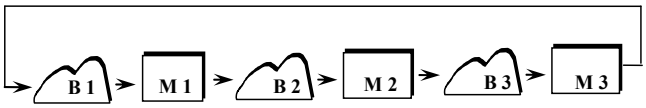
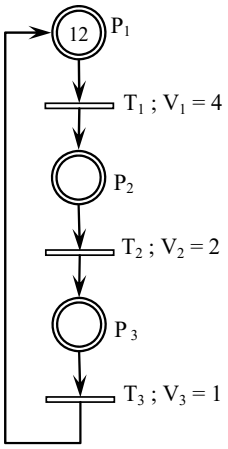


Figure 2 CCPN model of the manufacturing system



In a CCPN, the marking is no longer an integer but a real number. This approximation makes it possible to considerably reduce the number of events considered by the model. This facilitates the simulation and analysis of the behaviour of the model. This approximation has been largely dealt with in the papers presenting the continuous PNs (Alla and David, 1998). The transitions are associated maximal firing speeds. A transition T_j may be either:

1 strongly enabled, if all its input places are marked

2 weakly enabled, if all its input places which are empty are fed by firing of other transitions

3 not enabled, if some of its input places are neither marked nor fed by firing of other transitions.

Let us call $v_j(t)$ the instantaneous transition firing speed of transition T_j. It is equal to the maximal firing speed V_j if the transition is strongly enabled. In case of weakly enabled transition, the instantaneous transition firing speed is less than its maximal value but remains constant.

In the CCPN in Figure 2, if the initial marking transition T_1 is strongly enabled then it is fired with its maximal speed $v_1(0) = V_1 = 4$. If transitions *T_2 is weakly enabled, it is fed by a speed greater than its maximal value, then* $v_2(0) = 2$. In

the same way, T_3 *is weakly enabled and fired with speed* $v_3(0) = 1$. Due to the continuous evolution, a transition with a zero marking input place is enabled and fired. This can constitute a surprising result for the PN community. The way the speeds are computed is formalised in Algorithm 1.

Algorithm 1 Firing speeds calculation

1 **Inputs:** A constant speed continuous Petri net and a time instant t;
2 **Output:** $v(t)$, the instantaneous firing speed vector at time **t**;
3 Set up C_1 the constraints corresponding to: $0 \le v_j(t) \le V_j$;
4 Set up C_2 the constraints based on the balance $B_i(t) \ge 0$ if $m_i = 0$;
5 $P_{ne}^{(1)} = \{P_i \mid m_i > 0\}$;
6 $T_{SF}^{(1)} = \{T_j \mid {}^\circ T_j \in P_{ne}^{(1)}\}$;
7 k ← 1;
8 **While** $T_{SF}^{(1)} \neq T_{SF}^{(k+1)}$ and $T_{SF}^{(k)} \neq T$ do
9 $P_{ne}^{(k+1)} \leftarrow P_{ne}^{(k)} \cup (T_{SF}^{(k)})^\circ$;
10 $T_{SF}^{(k+1)} \leftarrow \{T_j \mid {}^\circ T_j \in P_{ne}^{(k+1)}\}$;
11 k ← k + 1;
12 **End while**
13 $T_{SF} \leftarrow T_{SF}^{(k)}$;
14 Set up the constraints C_3: $v_j(t) = 0$ for Tj ∉ T_{SF}
15 Solve the linear programming problem: $J_1 = \sum_{T_j \in T} v_j(t)$ given C_1, C_2 and C_3;

The markings dynamics are calculated thanks to the marking balance formalised in Definition 2 (David and Alla, 2010). It corresponds to a linear piecewise trajectory. The instantaneous transition firing speeds remain constant until the enabling conditions change, this occurs when the marking of a place becomes zero.

Definition 2 (marking balance): The balance of the marking of a place P_i in a CCPN is the difference between the feeding $I_i(t)$ and the draining $O_i(t)$ speeds of P_i. It corresponds to the marking time derivative.

$$B_i(t) = \dot{m}_i(t) = I_i(t) + O_i(t)$$

With:

$$I_i(t) = \sum_{j=1}^{m} Post(P_i, T_j).v_j(t)$$

$$O_i(t) = \sum_{j=1}^{m} Pre(P_i, T_j).v_j(t)$$

□

As has been explained, the marking value zero or not zero is a fundamental element in the CCPN dynamics. It determines the values of the firing speeds; it is why we have introduced the macro-marking notion.

Definition 3 (macro-marking): The macro marking $\tilde{\boldsymbol{m}}$ of a CCPN is a binary vector whose dimension is n (number of places), and whose value at time **t** is:

$$\tilde{\boldsymbol{m}}_i(\boldsymbol{t}) = \begin{cases} \mathbf{1} & \text{if } m_i(\mathbf{t}) > 0 \text{ or } B_i > 0 \\ \mathbf{0} & \text{if } m_i(\mathbf{t}) = 0 \text{ and } B_i = 0 \end{cases}$$

if $\forall i \in \{1, 2, \ldots, n\}$, $\tilde{m}_i(\boldsymbol{t}) = 1$, we speak about a full macro-marking.

□

In the CCPN in Figure 2, if the three balance markings of places P_1, P_2 and P_3 are respectively $B_1(0) = -3$, $B_2(0) = 2$ and $B_3(0) = 1$, then the initial macro-marking is a full macro-marking $\tilde{\boldsymbol{m}} = (1, 1, 1)$.

Property 1: With each macro-marking is associated a constant instantaneous firing speed vector.

□

The proof of this property is obvious since the enabling conditions of a transition depend only on the marked or not marked input places of this transition.

An algorithm permitting the calculation of the transition firing speeds using the macro-marking is given in David and Alla (2010). The idea of this algorithm is to compute the greatest speeds verifying constraints using linear programming. It can be used for any PN structure including conflicts; in that case solving strategies must be chosen. We summarise this algorithm here after. The following notations are used in the algorithm here after:

P_{ne} is the set of non-empty places (i.e., marked or fed places)

T_{SF} is the set of surely firable transitions.

An invariant behaviour state (IB-state), defined below, is a time interval where the macro-marking is constant and therefore the firing speeds are constant. This concept allows building the CCPN dynamics with a finite graph, while the reachable state space is infinite.

Definition 4 (IB-state): In a CCPN, an IB-state is the time interval where:

- $\tilde{\boldsymbol{m}}$, the macro marking is constant
- **v**, the transitions firing speed vector is constant.

□

The state of a CCPN at time $\boldsymbol{t}$ is given by its marking. The classical way to describe the behaviour of a CCPN is the evolution graph. It is a graph where each vertex corresponds to an IB-state, and where each arc corresponds to an event that changes the IB-state. Only one event may change the IB-state of a CCPN, i.e., the fact that a place marking becomes zero. It has been proved that the number of nodes of an evolution graph is finite if there is no conflict in the PN or if they are solved in a deterministic way (David and Alla, 2010). Figure 3 represents the evolution graph of the

CCPN of Figure 2. The behaviour of this CCPN comprises three IB-states. In the final IB-State, the three speeds are equal, the slowest machine imposes its speed to the other ones (i.e., $\mathbf{v(0)} = (1, 1, 1)$).

Figure 3 The evolution graph of the manufacturing system

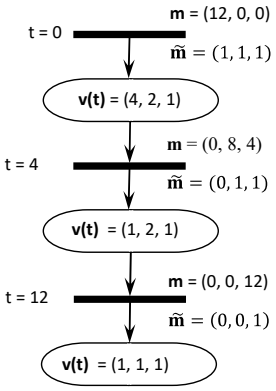
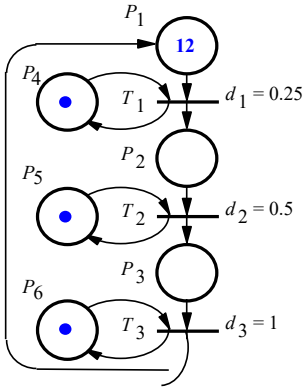
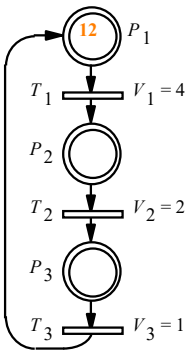


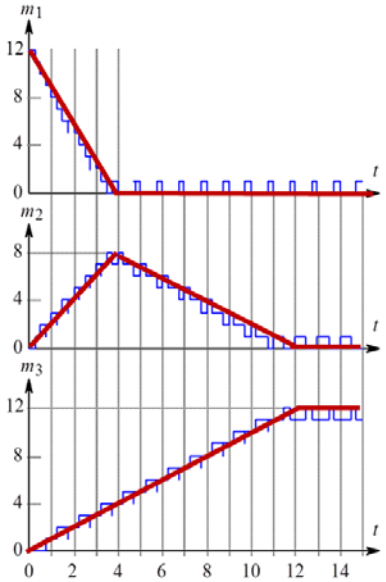
Figure 4 The manufacturing system, (a) CCPN model (in red), (b) corresponding discrete PN (in blue), (c) simulation of the discrete PN (exact discrete behaviour in red) and of the CCPN (approximate continuous behaviour, in blue) (see online version for colours)



(a)



(b)



(c)

Figures 4(a) and 4(b) show the CCPN (in red) of Figure 2 and the corresponding discrete PN (in blue) with an integer initial marking. This latter is obtained by replacing each maximal firing speed V_j by a duration $d_j = 1 / V_j$. Figure 4(c) is given the simulations of the discrete PN (exact behaviour) and of the CCPN (approximation). We can remark that the approximation given by the CCPN is satisfactory. The simulation duration is very fast; only two events dates have to be computed: occurrences of zero marking of places $P_1(t = 4)$ and $P_2(t = 12)$. This number of two remains the same whatever the number of pallets in the system, while for the discrete model, the simulation duration increases with the number of pallets.

3 Translation algorithm

An important contribution of this paper is the translation algorithm from a CCPN into a HA and its correctness proof. The idea of associating PNs and automata is not new. PNs are an easy-to-use modelling tool while automata have a great analysis power. This means that several researchers have associated them for coupling their advantages. One of the first works associates stochastic PN and Markov chains for the analysis of DESs (Frehse, 2005). For example, from the computation of the steady state of the Markov chain, it is possible to compute the performances of the PN model (mean markings and mean firings frequencies).

A first algorithm linking hybrid PNs with a hybrid automaton has been established in Allam and Alla (1998). This translation was done in a manual way. It is a kind of simulation, where at each step, the algorithm seeks the new occurring discrete event. Moreover, there is no proof of correctness of the obtained hybrid automaton. Our contribution is completely different from that proposed in Recalde et al. (2007) for two reasons:

- We propose a structural construction of the HA. It is independent from the initial marking, thanks to the concept of macro-marking. Given an initial marking, a unique HA is built and the actual model is obtained by marking the initial location.
- We formally prove that the HPN is bisimilar to the HA. That means that all the properties that are true for HPN are also true for the HA, and vice-versa.

We first present the HA in an intuitive way, a formal definition will be given later. A HA is composed by locations containing the continuous dynamics, and by arcs between these locations corresponding to the discrete transitions. The continuous variables are the places markings of the CCPN. In each location of the HA, the variables dynamics (the marking derivative) are the marking balance of each place. It is a constant real number. We associate with the initial location of the HA a full macro marking, i.e., the places are marked. Since the macro-marking vector is decreasing with time as shown in Figure 3, then it covers all the possible initial markings (David and Alla, 2010). Then, the translation algorithm is

structural, it is computed only once. Starting from a full macro-marking we can cover all the states that the CCPN can (structurally) reach. Since the only event that can change the dynamics is that a place becomes empty and then, this is only possible if its marking derivative is negative. We create then an output transition from the automaton location for each negative marking dynamics.

Algorithm 2 Translation of CCPN into HA

1 Input**:** A CCPN;
2 **Output**: A linear hybrid automaton;
3 **X**: a real valued vector of dimension n (le number of places of the CCPN);
4 Create L_0 the initial location;
5 Consider $\tilde{\mathbf{m}}^{\mathbf{0}}$ a full macro marking and calculate the firing speed using Algorithm 1, and thereafter the balance $\mathbf{B} = \dot{\mathbf{m}}$;
6 $\mathbf{M} \leftarrow \{\tilde{\mathbf{m}}^{\mathbf{0}}\}$; $\mathbf{L} \leftarrow \{L_0\}$; $l \leftarrow 0$; $j \leftarrow 0$;
7 $\dot{\mathbf{X}} \leftarrow \mathbf{B}$ for location L_0;
8 Associate to L_0 the invariant:
$$\bigwedge_{i|B_i<0} (m_i \geq 0)$$
9 **For** all L_k in L **do**
10 **For** all $i \mid B_i < 0$ in location L_k **do**
11 $l \leftarrow l + 1$;
12 $\tilde{m}_r^l \leftarrow \begin{cases} \tilde{m}_r^{l-1} & \text{for } r \neq i \\ 0 & \text{for } r = i \end{cases}$
13 **If** $\tilde{\mathbf{m}}^l \notin \mathrm{M}$ **then**
14 $\mathrm{M} \leftarrow \mathrm{M} \cup \{\tilde{\mathbf{m}}^l\}$
15 $j = j + 1$
16 Create a location L_j, and calculate the firing speed using Algorithm 1 and macro marking $\tilde{\mathbf{m}}^{\mathbf{1}}$, and thereafter the balance **B**;
17 $\mathrm{L} \leftarrow \mathrm{L} \cup \{L_i\}$;
18 Associate to L_j the Invariant:
$$\bigwedge_{s|B_s<0} (m_s \geq 0)$$
19 Create a transition T_{kj} from L_k to L_j, Set its guard ($m_i = 0$);
20 **Else**
Let L_r be the location associated to $\tilde{\mathbf{m}}^l$
21 Create a transition T_{kr} from L_k to L_r, Set its guard ($m_i = 0$);
22 **End if**
23 **End For**;
24 Define the initial state of the HA as the initial marking of the CCPN;
25 **End For**;

The HA obtained from the translation algorithm has the form of a tree since the number of IB-states is finite. The choice of the initial marking determines the reachable branch of the tree.

The hybrid automaton schematised in Figure 5 is the HA resulting from applying Algorithm 2 to the CCPN in Figure 2. The translation algorithm gives an automaton that has particular properties since we have particular dynamics and transitions guards, this is detailed in the following definition. We will denote it as continuous hybrid automata (CHA). For a general definition of a linear hybrid automaton, the reader may refer to Alur et al. (1995).

Definition 5 (Continuous hybrid automaton):

A continuous hybrid automaton CHA = (**L**, X, **δ**, ***F***, ***Inv***) such that:

1 **L** = $\{L_0, L_1, \ldots\}$ is a finite set of locations,
2 **X** = is the continuous state space. It corresponds to the vector $(x_1, x_2, \ldots, x_n)^T$ is the vector of n real-valued variables modelling the places marking; A valuation is a function that assigns a real-valued $\boldsymbol{u}(x_i) \in \Re^n$ to each variable x_i from **X**.
3 **δ** is a finite set of transitions, each transition is a 3-tuple $T = (L_k, g, L_l)$ such that:
- $L_k \in$ **L** is the source location;
- g is the transition guard, it is of the form: $m_i = 0$.
- $L_k \in$ **L** the target location;

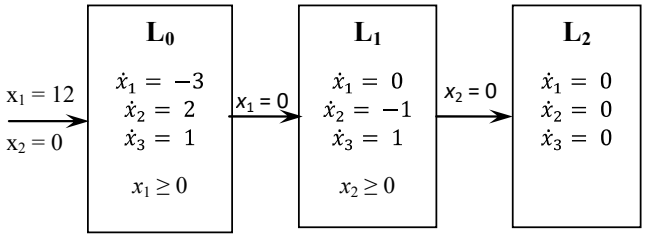
4 **F** is a function that assigns to each location a continuous linear vector field on **X**. While in discrete location L_k the continuous variable x_i evolve according to a differential equation of the form $\dot{x}_i = B_i$, where B_i is a the dynamic balance of the place P_i.
5 ***Inv*** is a function that affects to each location L_i, a linear predicate ***Inv***(L_k) of the form $\bigwedge_i x_i \geq 0$ that must be satisfied by the continuous variables in order to stay in location L_i.

□

In the sequel, we will express formally the semantics of a CCPN and of a CHA in term of timed transition systems. The state of a CCPN at time **t** is given by its marking vector **m**. It contains the sufficient information such that its knowledge at a time **t** allows the computation of the state at time **t** + dt. The state change may occur in two different manners:

1 in a discrete way, when a place marking becomes zero
2 continuously by time elapsing.

Figure 5 The hybrid automaton resulting from the translation algorithm



Definition 6 (semantics of a CCPN):

The semantics of a CCPN, *PN* = (**P**, **T**, *Pre*, *Post*, *V*, $\mathbf{m_0}$) is a timed transition system *ST* = (**Q**, q_0, →)) so that:

- **Q** = **m**, is the set of states
- $q_0 = \mathbf{m_0}$ is the initial state
- $\rightarrow \in \{m_i = 0\} \cup \Re^+$ is a discrete or a continuous transition, such that:
  1 The discrete transition corresponds to the emptying of a continuous place P_i.

$$\mathbf{m} \xrightarrow{m_i=0} \mathbf{m}' \text{ if:}$$

$$\begin{cases} v' \text{ is calculated using Algorithm 1} \\ \mathbf{m}' = \mathbf{m} \\ \forall k \neq i, \tilde{m}'_k = \tilde{m}_k \\ \tilde{m}_i = 1 \\ \tilde{m}'_i = 0 \end{cases}$$

  2 The continuous transition corresponds to the elapsing of time duration d;

$$\mathbf{m} \xrightarrow{e(d)} \mathbf{m}' \text{ if:}$$

$$\begin{cases} v' = v \\ \mathbf{m}' = \mathbf{m} + \mathbf{B}.\mathrm{d} \\ \tilde{\mathbf{m}}' = \tilde{\mathbf{m}} \end{cases}$$

□

B is the marking balance vector defined in Definition 3.

We will present now the semantics of a CHA.

Definition 7 (Semantics of CHA): The semantics of a continuous hybrid automaton is a timed transition system S_T = (**L**, L_0, →) so that:

- **Q** = (**L**, **X**) is the set of states
- $q_0 = (L_0, x_0)$ is the initial state
- $\rightarrow \in \boldsymbol{\delta} \cup \Re^+$ is the set of transitions that can be discrete or continuous.
  1 The discrete transition corresponds to the firing of transition T_j whose guard is if the form $x_i = 0$. This corresponds in the CCPN to the emptiness of a place marking;

$$(L_i, x) \xrightarrow{T_j} (L_{i+1}, x') \text{ if :}$$

$$\begin{cases} \exists (L_i, g, L_{i+1}) \in \boldsymbol{\delta} \text{ so that :} \\ g(x) = [x_i = 0],\ x' = x \\ Inv(L_{i+1})(x) = true \end{cases}$$

  2 Continuous transition corresponds to the elapsing of a time duration d;

$$(L_i, x) \xrightarrow{e(d)} (L_{i+1} x') \text{ if :}$$

$$\begin{cases} L_i + L_{i+1} \\ x' = x + B.d \\ \forall 0 \leq d' \leq d,\ Inv(L_i)(x + d') = true \end{cases}$$

□

In order to prove the correctness of the translation algorithm we define hereafter the equivalence between the states of CCPN and CHA.

Definition 8 (state equivalence ≈): Let **m** and (L_i, ***x***) be respectively a state of a CCPN and a state a CHA, then:

$$\mathbf{m} \approx (L_i, \mathbf{x}) \text{ if } \mathbf{m} = \mathbf{x}.$$

□

Any location of the CHA is determined by the value of the instantaneous firing speed vector (or by the macro-marking) of the CCPN. The continuous component of the automaton is given by the continuous marking of the CCPN.

4 Correctness of the translation algorithm

We prove in this section the correctness of the translation algorithm. We will show that the translation preserves the behaviour of the initial CCPN, in the sense that the semantics of the CCPN and the CHA are time-bisimilar. This proof is necessary since the analysis and the deduced properties are detremined from the CHA and remain true for the CCPN.

Definition 9 (Time bisimilarity): Let us consider two timed transition systems $S_1 = (Q_1, q_0^1, \rightarrow_1)$ and $S_2 = (Q_2, q_0^2, \rightarrow_2)$; and consider **R** a binary relation on $Q_1 \times Q_2$. The relation **R** is a strong bisimulation if:

$q_1 \rightarrow_1 q_1'$ and $q_1 \mathbf{R} q_2$, then $\exists q_2 \rightarrow_2 q_2'$ such that $q_1' \mathbf{R} q_2'$

$q_2 \rightarrow_2 q_2'$ and $q_1 \mathbf{R} q_2$, then $\exists q_1 \rightarrow_1 q_1'$ such that $q_1' \mathbf{R} q_2'$

□

In a time bisimilarity, if two states are equivalent then the firing of any type of transition preserves this equivalence. Let us call, SPN the semantics of the CCPN and S_{CHA} the semantics of its translation into a CHA.

Theorem 1 (Timed bisimilarity): A CCPN is temporally bisimilar to a CHA:

For all state $\mathbf{m} \in S_{PN}$ and $(L_i, \mathbf{x}) \in S_{CHA}$ such that $(L_i, \mathbf{x}) \approx \mathbf{m}$:

Firing of a discrete transition $m_i = 0$.

$$\mathbf{m} \xrightarrow{m_i=0} \mathbf{m}' \text{ iff } \exists (L_i, \mathbf{x}) \xrightarrow{m_i=0} (L_{i+1}, \mathbf{x}') \text{ and } (L_{i+1}, \mathbf{x}') \approx \mathbf{m}' \quad (1)$$

Firing of a continuous transition (time elapsing)

$$\mathbf{m} \xrightarrow{e(d)} \mathbf{m}' \text{ iff } \exists (L_i, \mathbf{x}) \xrightarrow{e(d)} (L_{i+1}, \mathbf{x}') \text{ and } (L_{i+1}, \mathbf{x}') \approx \mathbf{m}' \quad (2)$$

□

Proof:

Let us start by proof formula (1). Suppose that $(L_i, \mathbf{x}) \approx \mathbf{m}$, this mean, according to equivalence definition (Definition 8), that $\mathbf{m} = \mathbf{x}$, and that $\dot{\boldsymbol{m}} = \dot{\boldsymbol{x}}$ (this is true in CCPN since that the macro marking determines the marking dynamics). So if the i^{th} component of vectors $\mathbf{m}$ and $\mathbf{x}$ just become equal to zero, this will lead the CCPN to $\mathbf{m}'$ and the CHA to $(L_{i+1}, \mathbf{x}')$ where $\mathbf{m}' = \mathbf{x}'$.

For formula (2), suppose that $(L_i, \mathbf{x}) \approx \mathbf{m}$, the elapsing of a time duration d will lead the CCPN to the state $\mathbf{m}'$ such that $(\forall \boldsymbol{i}, m_i' = m_i + d)$ and the CHA to the state $(L_{i+1}, \mathbf{x}')$ such that $(\forall \boldsymbol{i}, x_i' = x_i + d)$. Since $\mathbf{m} = \mathbf{x}$, then $\mathbf{m}' = \mathbf{x}'$.

5 A water supply system

Consider a water supply system composed by three tanks used to feed consumers (Figure 6). Tank 1 is the main reservoir; it is fed either from natural sources with a water flow of 20 m^3/h, or from ground water with a water flow of 30 m^3/h. It supplies the two other tanks via pipes with maximal flow capacity 40 m^3/h. Tanks 2 and 3 are respectively linked with consumers 1, 2, 3 and 4, 5. The three tanks have limited respective capacities: 3,000 m^3, 1,200 m^3 and 1,000 m^3. When tank 1 is full the pumping rate is changed, while for tanks 2 and 3, the pipes input flow is reduced according to the output flow. The consumers' water demand is respectively 11, 8, 5 m^3/h for tank 2, and 7, 9 m^3/h for tank 3. Initially, the tanks are assumed empty. Our objective is to model this system by a CCPN, to translate it into a HA and to analyse its dynamic behaviour. Particularly, it will be seen if the demand is satisfied in time.

The water supply system is modelled by the CCPN in Figure 7. It is an exact model, where three structural conflicts appear; then resolution strategies should be given. We suppose that the three conflicts T_1-T_2, T_3-T_7 and T_4-T_5-T_6, are solved by priorities:

1 $T_1 < T_2$, T_1 has priority over T_2, (the water coming from the natural sources are used in priority)

2 $T_3 < T_7$

3 $T_4 < T_5 < T_6$.

Figure 6 A water supply system

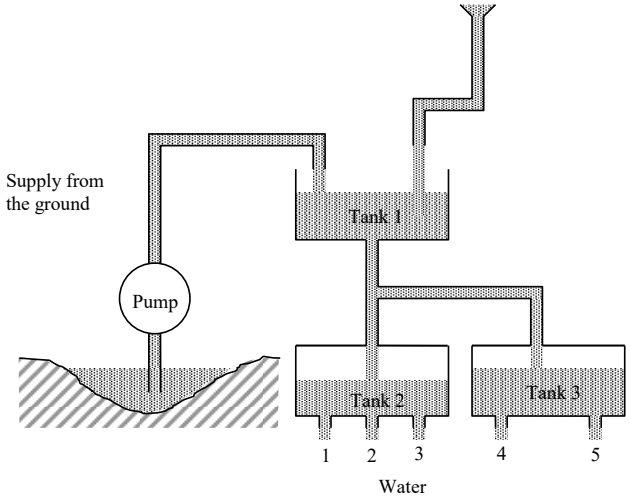
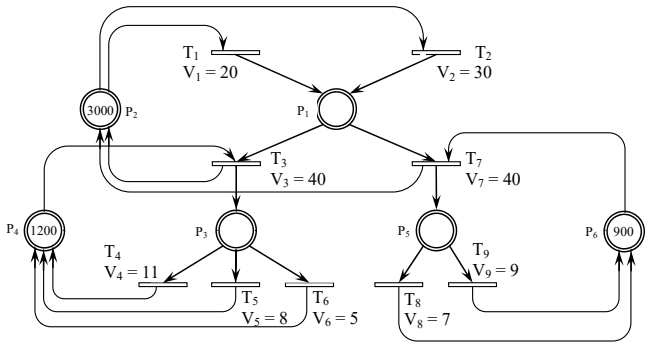


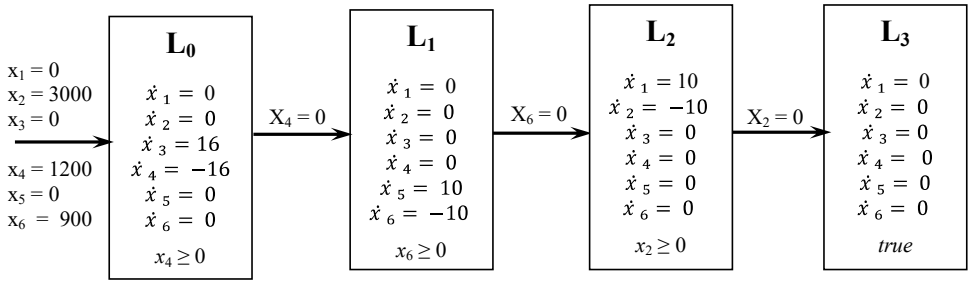
Figure 7 CCPN model of the water supply system



The conflict between T_8, T_9 is solved by sharing proportionally to their maximal speeds, $[T_8, T_9]$. Of course these conflicting policies can be changed. It is an intuitive model since the physical entities appear clearly, each tank is modelled by two places, which marking corresponds to the liquid volume in m^3.

The continuous hybrid automaton of the water supply system is obtained thanks to the translation algorithm. It is obvious that this model is less readable than the CCPN. Figure 8(a) gives the continuous hybrid automaton (Algorithm 2) and Figure 8(b) gives the instantaneous firing speed for each location (Algorithm 1). We can remark that in the last location the pumping rate (speed $v_2(t)$) decreases to 20 m^3/h since all the tanks are full. Figure 9 gives the physical state of the water supply system corresponding to each location; it allows illustrating the close link existing between the model and the physical process.

Figure 8 (a) The continuous hybrid automaton of the water supply system (b) The instantaneous firing speeds in each location

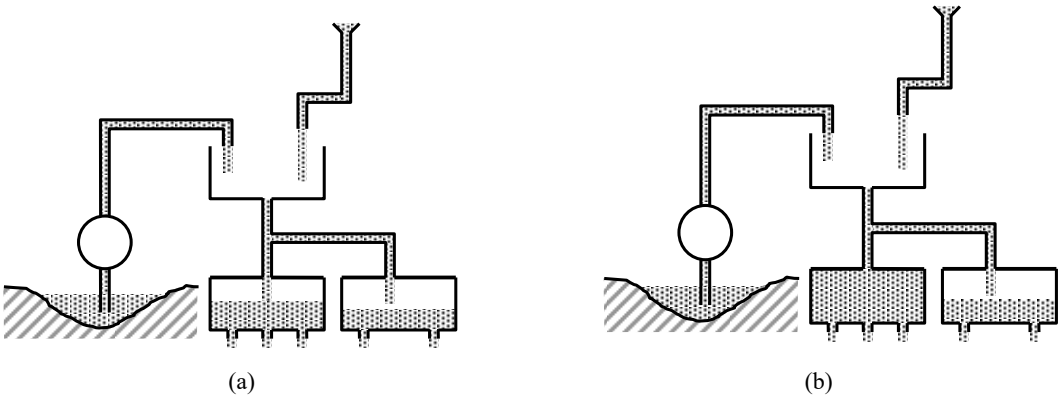


(a)

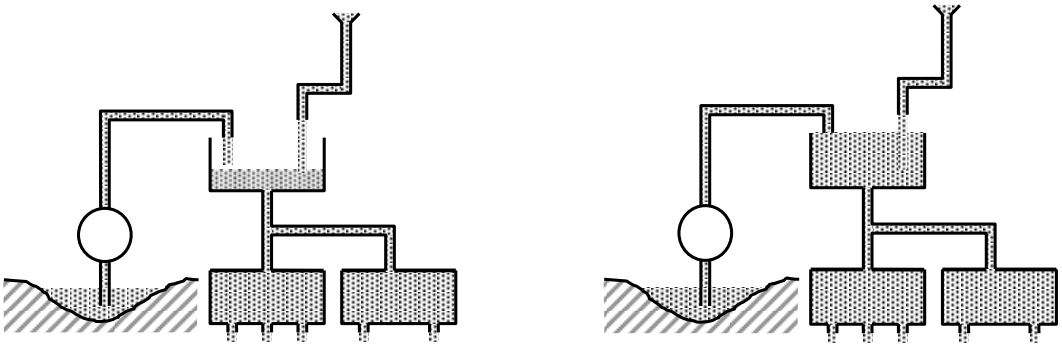
	L_0	L_1	L_2	L_3
$v_1(t)$	20	20	20	20
$v_2(t)$	30	30	30	20
$v_3(t)$	40	24	24	24
$v_4(t)$	11	11	11	11
$v_5(t)$	8	8	8	8
$v_6(t)$	5	5	5	5
$v_7(t)$	10	26	16	16
$v_8(t)$	4.375	7	7	7

(b)

Figure 9 The water supply system situation in each location, (a) in L_0 (b) in L_1 (c) in L_2 (d) in L_3



(a) (b)



(c) (d)

Figure 10 The reachable state space of the hybrid automaton of Figure 8(a)

$$Reachable state\ space = \begin{cases} L_0 \& \begin{cases} x_6 = 900 \\ x_5 = 0 \\ x_3 + x_4 = 1200 \\ x_2 = 3000 \\ x_1 = 0 \\ x_3 \geq 0 \\ x_3 \leq 1200 \end{cases} \\ L_1 \& \begin{cases} x_5 + x_6 = 900 \\ x_4 = 0 \\ x_3 = 1200 \\ x_2 = 3000 \\ x_1 = 0 \\ x_5 \leq 900 \\ x_5 \geq 0 \end{cases} \\ L_2 \& \begin{cases} x_6 = 0 \\ x_5 = 900 \\ x_4 = 0 \\ x_3 = 1200 \\ x_1 + x_2 = 3000 \\ x_1 \leq 3000 \\ x_1 \geq 0 \end{cases} \\ L_3 \& \begin{cases} x_6 = 0 \\ x_5 = 900 \\ x_4 = 0 \\ x_3 = 1200 \\ x_2 = 0 \\ x_1 = 3000 \end{cases} \end{cases}$$

It is now possible to draw properties of the CCPN model from the CHA. PHaVer (Ajmone-Marsan et al., 1989) is a software that allows the formal calculation of the reachable state space when the hybrid automaton is linear or rectangular in view of analysing its behaviour. From the CCPN, a linear HA has been obtained since the time derivatives are constant, and the reachable state space of the CHA of Figure 8(a) is given in Figure 10. From this analytical characterisation, it is possible to evaluate the performances of the water supply system such as the duration in each location or the maximal, minimal and average values of the contents of each tank. These performances can be obtained from the linear inequalities using optimisation solvers. For example, we can compute the duration of each phase corresponding to the sojourn time in each location. We then obtain 7.5 h, 90 h, 300 h for the first three locations; the last one has an infinite duration. From Figure 8(b), we can conclude that the demand is always satisfied except in location L_0 where the instantaneous firing speeds of transitions T_8, and T_9 are less than their maximal speeds.

We can remark that in the final location the water flows and the tanks levels are constant, till a modification for example of the demand or the flow rate of the natural sources. In order to have variable demands and flows, it is necessary to associate discrete behaviour with the continuous one; it is why hybrid PNs have been defined (David and Alla, 2010). However, the CCPN model constitutes the core of the hybrid model, and all the basic ideas presented in this paper will be directly introduced in the hybrid model.

6 Discussion

The fluidification of DESs is a concept that has been introduced in PNs (David and Alla, 2010). It opened the way for a multitude of works based on this idea. Particularly, important research exists in the field of biological systems such as the works in Doi et al. (2004) and Troncale et al. (2009). For these systems, continuous PNs have brought readable graphical models showing clearly the continuous dynamics in terms of continuous flows. Another interesting case study is constituted by a gas storage system controlled by valves, where the objective is to determine the gas throughput (Champagnat et al., 1998). The difficulty in that case is the nonlinear dynamics of the compressible gas. For both systems given above, the system is intrinsically continuous. Another important application domain is found in systems where large numbers of discrete entities are moving in short delays. A continuous flow can then approximate this behaviour. This occurs mainly for manufacturing systems (Demongodin and Giua, 2014) and transport systems (Dotoli et al., 2009). Our contribution can help for the modelling and analysis of all the types of cited systems, taking benefits of the graphical modelling of continuous PNs combined with the analytical analysis power of HA.

7 Conclusions

In this paper, we have studied the continuous PNs, where the maximal firing speeds are constant. We have presented the main features of its modelling and we have proposed for this class of PNs an algorithm that permits the translation into a hybrid automaton. This algorithm allows associating the powerful modelling of PNs with the powerful analysis of HA. We also proved the temporal bisimulation between the CCPN and the resulting hybrid automaton, thus guaranteeing the soundness of the translation. It is independent of the initial marking. The results were illustrated thanks to a water supply system.

An important objective of this algorithm is to use CCPN and the translated CHA for the analysis of hybrid dynamic systems and particularly for the synthesis of controller for positive systems. The results of this paper must be generalised to hybrid PN providing a structural translation with a formal proof of the correctness. Formal analysis of the obtained HA will allow a formal characterisation of the reachable state spaces, and then a formal computation of a controller or a diagnoser. This corresponds to our current research.

References

Ajmone-Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G. and Cumati, A. (1989) 'The effect of execution policies on the semantics and analysis of stochastic Petri nets for the performance of multiprocessor systems', *IEEE Trans. on Software Engineering*, Vol. 15, No. 7, pp.832–846.

Alla, H. and David, R. (1998) 'Continuous and hybrid Petri nets', *Journal of Circuits, Systems, and Computers*, Vol. 8, No. 1, pp.159–188.

Allam, M. and Alla, H. (1998) 'From hybrid Petri nets to automata', *Journal Européen des Systèmes Automatisés*, Vol. 32, Nos. 9–10, pp.1165–1185.

Alur, R. and Dill, D.L. (1994) 'A theory of timed automata', *Theoretical Computer Science*, Vol. 126, No. 2, pp.183–235.

Alur, R., Courcoubetis, C., Halbwchs, N., Henzinger, T.A., Ho, P.H., Nicolin, X., Olivero, A., Sifakis, J. and Yovine, S. (1995) 'The algorithmic analysis of hybrid systems', *Theoretical Computer Science*, Vol. 138, No. 1, pp.3–34.

Balduzzi, F., Giua, A. and Menga, G. (2000) 'First-order hybrid Petri nets: a model for optimization and control', *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 4, pp.382–399.

Brinkman, P.L. and Blaauboer, W.A. (1990) 'Timed continuous Petri nets: a tool for analysis and simulation of discrete event systems', *Proc. 1990 European Simulation Symposium*, Ghent (BE), November, pp.164–168.

Champagnat, R., Pingaud, H., Alla, H., Valentin-Roubinet, C., Flauss, J-M. and Valette, R. (1998) 'A gas storage example as a benchmark for hybrid modeling', *European Journal of Automation*, Vol. 32, Nos. 9–10, pp.1233–1253.

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 (FR), June, pp.217–225.

Cousot, P. and Cousot, R. (2010) 'A gentle introduction to formal verification of computer systems by abstract interpretation', in Esparza, J., Grumberg, O. and Broy, M. (Eds.): *Logics and Languages for Reliability and Security*, NATO Science Series III: Computer and Systems Sciences, IOS Press, pp.1–29.

David, R. and Alla, H. (2010) *Discrete, Continuous, and Hybrid Petri Nets*, February, Springer, Heidelberg Germany.

Demongodin, I. and Giua, A. (2014) 'IDynamics and steady state analysis of controlled generalized batches Petri nets', *Nonlinear Analysis Hybrid Systems*, May, Vol. 12, pp.33–49, DOI: 10.1016/j.nahs.2013.11.010.

Demongodin, I. and Koussoulas, N.T. (1998) 'Differential Petri nets: representing continuous systems in a discrete event world', *IEEE Transactions on Automatic Control*, Vol. 38, No. 4, pp.573–579.

Demongodin, I. and Koussoulas, N.T. (2008) 'Differential Petri net models for industrial automation and supervisory control', *IEEE Trans. on Systems, Man, and Cybernetics*, July, Vol. 36, No. 4, pp.543–553.

Doi, I.A., Fujita, S., Matsuno, H., Nagasaki, M. and Miyano, S. (2004) 'Constructing biological pathway models with hybrid functional Petri nets', in *Silico Biology*, Vol. 4, No. 3, pp.271–291.

Dotoli, M., Fanti, M.P., Iacobellis, G. and Mangini, A.M. (2009) 'A first order hybrid petri net model for supply chain management', *IEEE Transactions on Automation Science and Engineering*, November, Vol. 6, No. 4, pp.744–758, ISSN: 1545-5955.

Frehse, G. (2005) 'PHAVer: algorithmic verification of hybrid systems past HyTech', *Proceedings of the Fifth International Workshop on Hybrid Systems: Computation and Control (HSCC), Lecture Notes in Computer Science*, Springer-Verlag, (extended and revised), Vol. 3414, pp.258–273.

Júlvez, J., Recalde, L. and Silva, M. (2006) 'Deadlock-freeness analysis of continuous mono-T-semiflow Petri nets', *IEEE Transactions on Automatic Control*, Vol. 51, No. 9, pp.1472–1481.

Kara, R., Loiseau, J-J. and Djennoune, S. (2008) 'Quantitative analysis of continuous weighted marked graphs', *Nonlinear Analysis: Hybrid Systems*, November, Vol. 2, No. 4, pp.1010–1020, Elsevier.

Lohmann, N., Verbeek, H.M.W. and Dijkman, R.M. (2009) 'Petri net transformations for business processes a survey', *Transactions on Petri Nets and Other Models of Concurrency: ToPNoC II, LNCS*, Vol. 5460, pp.46–63.

Mahulea, C., Recalde, L. and Silva, M. (2009) 'Basic server semantics an performance monotonicity of continuous Petri nets', *Discrete Event Dynamic Systems: Theory and Applications*, June, Vol. 19, No. 2, pp.189–212.

Norazura, A., Noraida, I.G., Anton, A.K. and Razman, M.T. (2015) 'Modelling the complexity of emergency department operations using hybrid simulation', *Int. J. of Simulation and Process Modelling*, Vol. 10, No. 4, pp.360–371.

Recalde, L. and Silva, M. (2001) 'PN fluidification revisited: semantics and steady state', *Journal Européen des Systèmes Automatisés*, Vol. 35, No. 4, pp.435–449.

Recalde, L., Haddad, S. and Silva, M. (2007) 'Continuous Petri nets: expressive power and decidability issues', in *Proc. of 5th Int. Sym. of Automated Technology of Verification and Analysis, LNCS*, Springer, Tokyo, Vol. 4762, pp.302–377.

Troncale, S., Comet, J-C. and Bernot, G. (2009) 'Enzymatic competition: modeling and verification with timed hybrid Petri nets', *Pattern Recognition*, Vol. 42, No. 4, pp.562–566.

van Dijk, N.M. (1999) 'On hybrid combination of queuing and simulation', *DSI'99, Decision Sciences Institute, 5th International Conference*, Athens, Greece, 4–7 July, pp.1131–1133.

Wainer, G.A. (2016) 'Real-time simulation of DEVS models in CD++', *Int. J. of Simulation and Process Modelling*, Vol. 11, No. 2, pp.138–153.