



HAL
open science

Fault-tolerant continuous flow systems modelling

B Tolbi, H Tebbikh, Hassane Alla

► **To cite this version:**

B Tolbi, H Tebbikh, Hassane Alla. Fault-tolerant continuous flow systems modelling. International Journal of Systems Science, 2017, 48 (1), pp.107-117. 10.1080/00207721.2016.1160454 . hal-01412952

HAL Id: hal-01412952

<https://hal.science/hal-01412952>

Submitted on 9 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ORIGINAL PAPER

Fault-tolerant continuous flow systems modelling

B. Tolbi^{a*}, H. Tebbikh^a and H. Alla^b^aLAIG, University of Guelma, B.P. 401, Guelma, Algeria; ^bGipsa-Lab, University of Joseph Fourier, Grenoble, France

(Received 00 Month 20XX; final version received 00 Month 20XX)

This paper presents a structural modelling of faults with Hybrid Petri Nets (HPNs) for the analysis of a particular class of hybrid dynamic systems, continuous flow systems. HPNs are first used for the behavioural description of continuous flow systems without faults. Then faults' modelling is considered using a structural method without having to rebuild the model to new. A translation method is given in hierarchical way, it gives a Hybrid Automata (HA) from an Elementary Hybrid Petri Net (EHPN). This translation preserves the behavioural semantics (timed bisimilarity), and reflects the temporal behaviour by giving semantics for each model in terms of Timed Transition Systems (TTS). Thus, advantages of the power modelling of HPNs and the analysis ability of HA are taken. A simple example is used to illustrate the ideas.

Keywords: Continuous flow systems; Elementary Hybrid Petri Nets; faults modelling; Hybrid Automata; Timed Transition Systems; Timed bisimilarity.

Index to information contained in this paper

1. Introduction
2. Faults modelling
 - 2.1. Hybrid Petri nets
 - 2.2. Hybrid automata
3. From elementary hybrid Petri nets to hybrid automata
 - 3.1. Structural translation
 - 3.2. Semantics of the two basic models
 - 3.3. Timed similarity and bisimilarity
4. Case study
5. Conclusion

1. Introduction

Hybrid dynamic systems (HDSs) are dynamic systems integrating explicitly and simultaneously continuous and discrete event systems, which require for their description the use of a continuous model, and a discrete event model (Pettersson, Lennartson, et al., 1995). In this paper, a particular class of hybrid dynamic systems, which are continuous flow systems is considered, for example transportation systems, production systems, communications systems...etc. This class involves hybrid systems, which are said positive, i.e. all the state variables take only positive values. In the literature, a particular effort was given to study this class of hybrid dynamic systems. Indeed, it is sufficiently rich to allow a realistic modelling of many actual problems. This assumption allows

*Corresponding author. Email: bilalcdsi@gmail.com

us to model these systems by HPNs that have the advantage of being a powerful model for HDSs description. The model is designed intuitively; keeping the main characteristics of PNs such as the explicit representation of parallelism, synchronization and conflicts.

Generally, faults' modelling involves the acquisition of a priori knowledge of faults to be detected, and to improve diagnostic techniques (detection, isolation and identification), it is necessary to have a strong and accurate faults modelling. This problem would still be the biggest problem for automation engineers, there are many works involved and carried in literature, modelling of faults with PN model (Renganathan & Bhaskar, 2011). Among them, in (Renganathan & Bhaskar, 2013), authors proposed an observer based technique to detect faults on the system modelled by HPNs. Another work focused on a novel fault diagnosis and cause analysis model using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets (Liu, Lin, & Ren, 2013). Others proposed new algorithms to achieve fault diagnosis and fault tolerant control on the model based on HPNs tool in (David & Alla, 2001). None of these models possesses a powerful analysis technique; they are mainly descriptive.

The hybrid PN formalism was defined in (David & Alla, 2001), it combines in the same model a continuous PN, which represents the continuous flow, and a discrete T-time PN (Ramchandani, 1974), to represent the discrete behaviour. Hybrid Automata (HA) constitute a powerful analysing tool. They were introduced by (Alur, Courcoubetis, Henzinger, & Ho, 1993), this tool can model the widest variety of continuous dynamics; it also allows a formal analysis of HDSs. HA are finite state automata, extended with real value variables which *evolve* according to differential equations; they are the most general HDSs formalism, since they can model a largest variety of HDSs.

HPNs do not require an exhaustive enumeration of the state space and may represent, with finite way, a system whose reachable space is infinite. It includes in the same formalism, the representation of discrete dynamical and discrete event behaviours. However, this makes the analysis task of the model very difficult. Therefore, *the idea is to couple* the analysis power of HA to the modelling power of HPNs. That is why a translation procedure has been introduced. This idea was already considered for Time Petri Nets (TPNs) (Sifakis & Yovine, 1996), PN with deadlines (Bornot, Sifakis, & Tripakis, 1998), T-time PN (Cassez & Roux, 2006), bounded T-time PN (Sava, 2001), Batches PN (Demongodin & Rouibia, 2003), and some classes of HPN (Allam & Alla, 1998; Cortes, Eles, & Peng, 2000). This is a difficult task for general HPNs, that is why Elementary HPNs (EHPNs) have been chosen, they separate the dynamics of the discrete part from the dynamics of the continuous part at a certain level (there are no marking transformation, from discrete to continuous or vice-versa). This model has a strong property since as it will be shown later, it allows a structural construction of the HA.

Allam and Alla presented in (Allam & Alla, 1998), an algorithm that allows the construction of the hybrid automaton equivalent to HPN. In (Cortes et al., 2000) the authors consider an extension of TPN and propose a translation into hybrid automata. In another line of work, (Sava, 2001) considers bounded TPN where the underlying Petri Net is not necessarily safe and proposes an algorithm to translate the TPN into a Timed Automaton (TA). Lime and Roux proposed an extension in (Lime & Roux, 2003) of the state class graph construction that allows to build the state class graph of a bounded TPN as a TA. They prove that this TA and the TPN are timed-bisimilar. Cassez and Roux proposed a structural translation in (Cassez & Roux, 2006) from TPN to TA, the correctness of translation is proved. In (Etouati, Yeddes, Hadj Alouane, & Alla, 2009) authors proposed a systematic method to convert Extended TPN to Linear Hybrid Automata (LHA) through a translation algorithm in order to analyze the behaviour and properties of the system. And among recent work, Ghomri presented in (Ghomri, 2012) an algorithm that allows the construction of the hybrid automaton equivalent to D-elementary HPN. A model which is an extension of this one will be used and a proof of the correctness of the translation will be established.

Our contribution. *In this work, faults modelling with HPNs will be studied by* a structural method; faults and normal behaviour are modelled separately. Firstly, an independent fault model

is introduced to represent independently faults. Faults modelling on discrete part will not be presented in this paper, only on continuous part. Secondly, a simple modelling technique will be done to describe system without faults. Finally, a composition of independent models will be presented to get the global model. **A powerful HPN model is necessary to show the faults consequences on the continuous part, and for behaviour analysis, HA model is also necessary. For this, formal semantics of EHPNs and HA will be given in terms of TTS and the analysis power of HA will be coupled to the modelling power of EHPNs using a structural translation from EHPN to HA that preserves the semantics (in the sense of timed bisimilarity) of both models. The technique of bisimilarity will be used to prove the correctness of our translation procedure by giving the proof of this one.**

Outline of the paper. Section 2 introduces faults modelling using the general model of HPNs, in this section the easy way of faults modelling will be presented. Section 3 explains our proposal procedure to translate the sub-HPN named EHPNs into HA. In this Section, also the way how to prove the bisimilarity between EHPNs and HA as TTS formalism will be showed. Section 4 provides a case study of our translation method. Finally, our ongoing work and perspectives will be given in Section 5.

2. Faults modelling

Let us note that a system fault corresponds to a dysfunction state, while a failure or a fault source of failure is an event that can lead to a state of dysfunction. In the context of HDSs, the occurrence of fault is, also, the passage towards a dysfunction state. This passage can be modelled by a transition on a fault, where one considers an event-based modelling (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzi, 1995). Other authors consider a state-based modelling (Zad, Kwong, & Wonham, 1999), an approach combining the advantages of state and event based modelling is proposed in (Sayed Mouchaweh, Philippot, & Carré-Ménétrier, 2008). A fault can also be represented as an execution of a given supervision pattern, which is a temporal property related to the occurrence of a set of trajectories/events that must be diagnosed (Jeron, Marchand, Pinchinat, & Cordier, 2006). These techniques use models including faulty behaviours. There are also approaches, which use fault-free models; they are based on comparing the systems outputs with the models nominal outputs. The fault-free modelling approach proposed by (Pandalai & Holloway, 2000) uses condition templates to determine if the system generates events in the right order or within the given time delays. In (Sayed-Mouchaweh, 2012) expert knowledge is associated with condition templates to identify the faults related to missing or unexpected events, and progressive monitoring is used to reduce the set of fault candidates after the occurrence of new observable events. Another practical fault-free modelling approach for fault diagnosis of manufacturing systems has been proposed in (Roth, Lesage, & Litz, 2011).

2.1. Hybrid Petri nets

In this formalism, the firing of a continuous transition describes the material flow, while the firing of a discrete transition models the occurrence of an event that can for example change firing speeds of the continuous transitions.

The work presented in this paper is interested, and restricted, only to EHPNs formalism, since there is a decoupling between the discrete and the continuous parts (one part may influence the behaviour of the other one, but there is no transformation of discrete marking into continuous marking or vice-versa). They combine a time PN and a constant speed continuous PN (CCPN)(R.David & Alla, 2010). A formal presentation of EHPNs will be given in Definition 3.

This paper is based on some of techniques which are presented at the beginning of this section, but here is not needed to rebuild automaton model to new or build both automaton model (model

without faults and model with faults) (Derbel, Alla, Ben Hadj-Alouane, & Yeddes, 2009). **This work is aimed to build** the system complete model (model with faults), by modelling the system in normal functioning, and an independent modelling of faults. The model that will be used is HPN. In this model, the faults are considered as unobservable events, this leads to decompose the sets of events in two subsets: the set of observable events and the set of unobservable events (Σ_o and Σ_u). If the unobservable event is associated with a discrete transition, then this transition is said to be unobservable. This work can be extended into general HPN, but the translation algorithm will be difficult, mostly the continuous-discrete decoupling will be losed and therefore the structural character of translation. However, this problem could be studied as a future perspective.

Definition 1: A hybrid PN model in normal functioning is an 8-uplet $\mathbf{N}=(P, T, Pre, Post, h, \Sigma, I, V, M_0)$, such that:

1. $P=\{P_1, P_2, \dots, P_n\}$ is a finite set of n places. $P= P^C \cup P^D$;
2. $T=\{T_1, T_2, \dots, T_m\}$ is a finite set of m transitions. $T= T^C \cup T^D$;
 - o T^D is a finite set of discrete transitions;
 - o T^C is a finite set of continuous transitions;
3. $Pre : P \times T \rightarrow N$ and $Post : P \times T \rightarrow N$ are the backward and forward incidence mappings; These mappings are such that:
 $\forall (P_i, T_j) \in P^D \times P^C, pre(P_i, T_j) = post(P_i, T_j)$;
4. $h : P \cup T \rightarrow \{C, D\}$, defines the set of continuous nodes ($h(x) = C$) and discrete nodes ($h(x) = D$);
5. Σ is a set of events; $\Sigma = \Sigma_o \cup \Sigma_u$
 - o Σ_o is a sub-set of observable events;
 - o Σ_u is a sub-set of unobservable events;
 - o $\Sigma \rightarrow Q^+ \times Q^+$ associates to each event σ_j , an occurrence interval $d_j = [\alpha_j \beta_j]$.
6. $I : T^D \rightarrow \Sigma$ associates an event to each D-transition T_j ;
7. $V : T^C \rightarrow R^+$ associates a maximal firing speed V_j to each C-transition T_j .
8. M_0 is the initial marking, such that $M=(m_C, m_D)^T$ m_C and m_D are the discrete and the continuous markings.

In order to illustrate the HPN modelling let us consider the following example.

Example 1: Let consider a road section that can tolerate a maximum number of cars 150 (assuming that average distance between two cars is $L=4m$). Assuming that the section has an entry ramp, the two section entries are regulated by a traffic light, **the green light turns on in the first and the second entry, respectively after $d_1 = [20\ 25]$ and $d_2 = [25\ 30]$** . Cars enter the section with an average speed of 30 km/h from both entries and leave it with an average speed of 48 km/h. The example is given in figure 1(a).

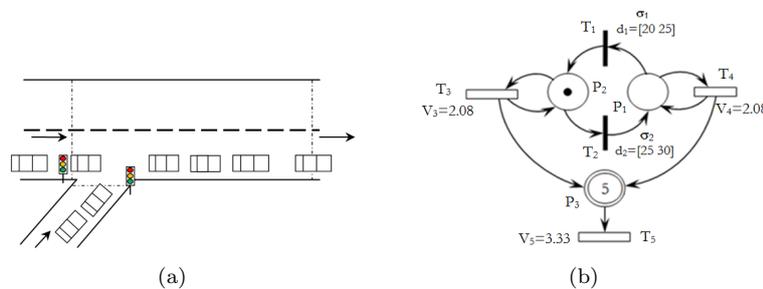


Figure 1. (a). Road section. (b) Hybrid PN of the normal functioning.

Figure 1(b) gives the HPN in normal functioning of the road section. The continuous part is represented by a double line, the continuous place P_5 represents the section and it is associated by its maximum number of cars (150) which are represented by tokens. The continuous transitions are associated by their maximal firing speeds "for example: $V_5 = (48km/h \times 1000)/(3600 \times 4m)$ ". The discrete part is represented by simple line where the transitions are associated by occurrence durations (d_1 and d_2) of events (σ_1 and σ_2) which correspond to the occurrence durations of green lights on the two section entries.

Our objective is to describe independently faults behaviour, representing just the faults. Faults modelling on the discrete part is not discussed here as it is widely studied. Only faults that have a consequence on the continuous part are presented. Fault occurring on the continuous aspect may be modelled by a continuous transition controlled by the occurrence of an unobservable event associated to a discrete transition. This is defined bellow.

Definition 2: A single fault model is a sub-hybrid PN \mathbf{F} which contains:

1. P_1, P_2 , two discrete places;
2. T_1 a discrete transition associated with fault σ_f (it is an unobservable event);
3. T_2 a continuous transition with constant firing speed V_2 ;
4. $Pre_f : P_2 \times T_2 \rightarrow N$ and $Post_f : P_2 \times T_2 \rightarrow N$ are the backward and forward incidence mappings of fault such that:

$$Pre_f(P_2, T_2) = Post_f(P_2, T_2) = 1;$$

5. $M(P_1) = 1, M(P_2) = 0$ is the initial marking.

Let us now complete example 1. A faulty situation of traffic can be observed if the green light is turned on, in both entries; the passage to this faulty state is due to the occurrence of the fault event (σ_f). The HPN model of this fault is given in Figure 2(a). It models the fault and its influence on the continuous behaviour. After occurrence of fault event, P_2 is marked and continuous transition T_2 is enabled at its maximal speed V_2 . This gives an additive flow to place P_3 , this behaviour is shown by the evolution graph defined in Figure 2(b). This evolution graph contains two states. The change from one state to another is assured at the fault occurrence, and that is reflected with firing speed modification of the continuous transition that becomes strictly positive. This change in dynamics will be useful in the diagnoser synthesis in our future work.

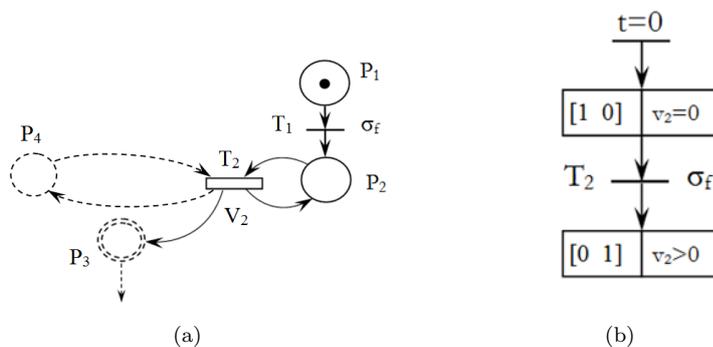


Figure 2. (a) Hybrid PN of the fault model of the road section. (b) Corresponding evolution graph.

Remark 1: It is possible to have a subtractive flow at the occurrence of the fault, for that we must reverse the arc that links transition T_2 and place P_3 .

To get the global model, it is needed to perform a composition of both models (normal functioning and fault models), this composition is structural and independent from the initial marking. And then the model with faults is translated directly to get an automaton model ready for analysis,

even in the presence of faults. The global model is defined below.

Definition 3: A global HPN model \mathbf{G} is a structure obtained from the composition of the models \mathbf{N} and \mathbf{F} (defined respectively in Definition 1 and 2). It is an 8-uplet $\mathbf{G}=(P^G, T^G, Pre_g, Post_g, h, \Sigma_g, I, V, M_0g)$, such that:

1. $P^G=\{P_1, P_2, \dots, P_{n+2}\}$ is a global finite set of $n + 2$ places (P_1 and P_2 are the two discrete places of \mathbf{F});
2. $T^G=\{T_1, T_2, \dots, T_{m+2}\}$ is a global finite set of $m + 2$ transitions (T_1 and T_2 is a fault discrete and continuous transition respectively of \mathbf{F});
3. $Pre_g, Post_g$ are the global backward and global forward incidence mappings;
4. $h : P \cup T \rightarrow \{C, D \}$, defines the set of continuous nodes ($h(x) = C$) and discrete nodes ($h(x) = D$);
5. Σ_g is the set of all events; $\Sigma_g=\Sigma_o \cup \Sigma_u + \{\sigma_f\}$;
6. $I : T^D \rightarrow \Sigma_g$ associates an event to each D-transition T_j ;
7. $V : T^C \rightarrow R^+$ associates a maximal firing speed V_j to each C-transition T_j .
8. $M_0g=[1 \ 0 \ M_0]^T$ is the global initial marking, (M_0 is the initial marking of \mathbf{F}).

Remark 2:

- o For multiplicative faults, the structure that corresponds to each type of fault must be duplicated;
- o A fault that corresponds to a structure change leads to a new HPN (this is not formalized in our paper).

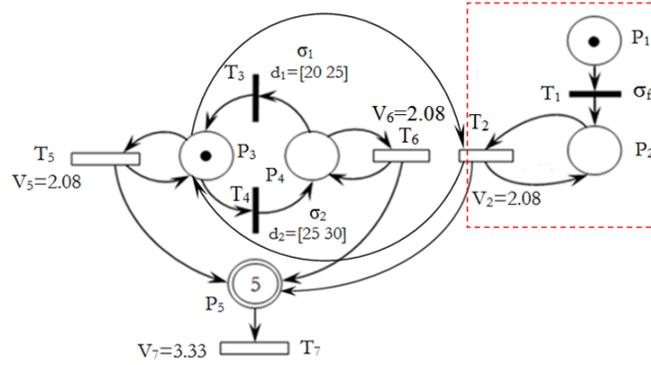


Figure 3. Global HPN model of the road section.

Figure 3 gives the global model of the road section.

2.2. Hybrid automata

To get the powerful model, it is necessary to couple the analysis power of HA to the modelling power of EHPN by preceding a structural translation from EHPN to HA which is defined below.

Definition 4: A hybrid automata is a 6-uplet $HA=(Loc, x, E, \delta, F, inv)$, such that:

1. Loc is a finite set of locations;
2. x is the continuous state space. x_C is the vector of real-valued variables modelling the continuous places marking and x_D is the vector of clocks corresponding to enabled transitions. A valuation is a function that assigns a real-valued $v(x) \in R$ to each variable $x_i \in x$.
3. E is a set of events;
4. δ is a finite set of transitions, each transition is a quintuple $T=(q, a, g, init, q')$ such that:

- $q \in Loc$ is the source location;
 - $a \in E$ is the event associated to the firing of T ;
 - g is the transition guard, it is a linear predicate on x ; a transition can be fired whenever its guard is satisfied;
 - $init$ is a reset function that affects a linear expression to variables of x when taking the corresponding transition;
 - $q' \in Loc$ is the target location;
5. F is a function that assigns to each location a continuous linear vector field on x . While in discrete location q , the continuous variables $m_i \in m_C$ evolve according to a differential equation of the form $\dot{m}_i = B_i$, where $B_i \in R^+$ is a the dynamic balance of the continuous place P_i , and the clocks $t_j \in x_D$ evolves according to the differential equation $\dot{t}_j = 1$;
6. inv is a function that affects to each location q , a linear predicate $inv(q)$ that must be satisfied by the continuous variables in order to stay in the location q .

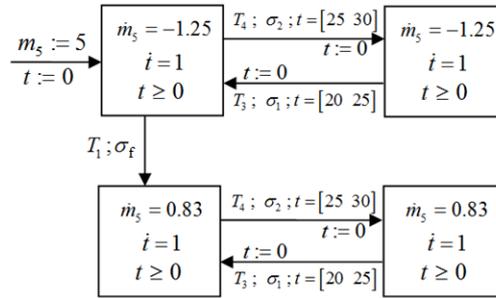


Figure 4. HA of the road section.

Figure 4 gives the HA of the road section, it is obtained thanks to Algorithm 1 given in Appendix 1. In each node, the time derivative of the continuous place corresponds to its balance marking (input flow minus output flow).

Most of works developed on HDSs return to a generalization of results coming from continuous systems (component approach), where few switching are considered. These approaches extend results of the continuous domain, such as unified models (Branicky, 1995), bond-graphs with switches (Buisson & Cormerais, 1998). The most considered faults are sensors faults, such as sensor offset or sensor stuck-off/stuck-on, which represent discrepancies between the measured and real values of system variables.

The approaches coming from DESs (Discrete Event Systems) consider the global behaviour of a process (system approach), where breaking dynamics are modelled. The main models are time Petri nets (Berthomieu & Menasche, 1983), timed automata (Alur & Dill, 1994), hybrid automata (Alur et al., 1995), hybrid Petri nets (Bail, Alla, & David, 1991), batch Petri nets (Demongodin, 2001). In these models the discrete part is complex while the continuous one is often simpler. The faults are boolean and lead to on/off behaviours. The major interest of our work is to consider the faults in the continuous part in an elegant way. Indeed, by modifying the balance marking of a continuous place in a HPN, it is possible to model a breakdown of a sensor or an additional flow due to a leakage or a feeding. This physical aspect of the fault description highlights the practical interest of this modelling.

3. From elementary hybrid Petri nets to hybrid automata

In this Section, hybrid automata are built from EHPNs so that the behaviours of both are in a one-to-one correspondence using the proposed algorithm. The HA resulting from the translation and the EHPN model will be proved to be timed bisimilar. To achieve this goal, formal semantics of HA and EHPNs are given in terms of TTS. Therefore, the bisimilarity between a TTS of an EHPN and a TTS of a HA is demonstrated. A bisimilarity between two transition systems ensures that any action of one of the systems can be simulated by the other (equivalent behaviours).

3.1. Structural translation

Our objective is to realize a structural translation from the sub-model EHPNs to HA. An EHPN is characterized by the following structure linking a continuous place to a discrete transition or a continuous transition to a discrete place (there is no transformation of tokens).

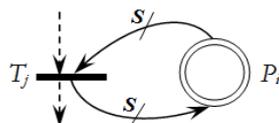


Figure 5. Mandatory structure in an EHPN.

Each marking in time PN will correspond to a CCPN configuration. An EHPN model can therefore be studied in a hierarchical manner, firstly the discrete part is considered, and then for each reachable discrete marking, the continuous configuration is considered. The translation process of this work is given in an informal way by the following steps:

- Step 1: Build the reachable marking graph of subjacent autonomous PN by isolating the time PN part from continuous part (CCPN) of hybrid model;
- Step 2: Associate respectively the dynamics¹ and the guards to each location and transition according to the continuous part configuration of hybrid model, where each reachable discrete marking correspond to a CCPN configuration and null flow vectors must be suppressed.
- Step 3: Build the final HA, where the value of the entry guard determines the target location whose downstream configuration of each location. The unreachable locations must be deleted.

The main part of our work concerns the structural modelling of faults by EHPN tool, which is a powerful tool for modelling. This contribution is not addressed in the paper (Ghomri, 2012), only the translation part is similar. Our translation procedure shows the effect of continuous place on discrete one and separates the dynamics of the discrete part from the continuous part at a certain level (there are no marking transformation, from discrete to continuous or vice-versa). In our case, the structure given in Figure 5 is added and for the first time, a mathematic proof for our proposed procedure is presented. This last will be detailed at the end of Section 3.3.

3.2. Semantics of the two basic models

TTS is used to describe the set of states and transitions between these states (Larsen, Pettersson, & Yi, 1995). Two types of transitions are possible in a TTS: a continuous transition or time transition that describes the flow of time or a continuous evolution and a discrete transition describing the

¹the marking balance corresponds to the difference between the input and the output flow for each continuous place P_i of a CCPN.

evolution following a discrete action (event). If an initial state is isolated in a TTS, the different paths starting from this state represent the different possible evolutions for modelled system. In this work, the emptying of a C-place (C1-event (R.David & Alla, 2010)) is considered like discrete transition. The semantics of HA and EHPNs are defined bellow.

Definition 5: Hybrid Automaton Semantics. The semantics of a hybrid automaton HA is defined as timed transition system $S_{HA} = (S^A, S_0^A, E, \longrightarrow_A)$ so that:

- $S^A = (q, x)$ is the set of states;
- $S_0^A = (q_0, x_0)$ is the initial state;
- E is the set of actions (events);
- $\longrightarrow_A \in S^A \times (E \cup R^+) \times S^A$ is the set of transitions that can be discrete or continuous, such that:

Discrete transition: firing transition T_j in the hybrid automaton.

$(q, x) \xrightarrow{T_j}_A (q', x')$ if:

$$\left\{ \begin{array}{l} \exists (q, a, g, \text{init}, q') \in \delta \text{ so that:} \\ g(x) = \text{true}, \text{inv}(q)(x) = \text{true} \\ \text{and } \text{inv}(q')(x') = \text{true} \end{array} \right.$$

Continuous transition: Time elapsing.

$(q, x) \xrightarrow{d}_A (q', x')$ if:

$$\left\{ \begin{array}{l} q' = q \\ x' = F(d) \\ \forall 0 \leq d' \leq d, \text{inv}(q)(x + d') = \text{true} \end{array} \right.$$

Definition 6: Elementary Hybrid PN Semantics. The semantic of an EHPN is a timed transition system $S_{EHPN} = (S^E, S_0^E, \Sigma_g, \longrightarrow_E)$ so that:

- $S^E = (m_D, m_C, t, v)$ is the set of states;
- $S_0^E = (m_{D0}, m_{C0}, \bar{0}, v_0)$ is the initial state;
- Σ_g is the set of actions (events), $\Sigma_g = T^D \cup \{m_i = 0\}$
- $\longrightarrow_E \in S^E \times (\Sigma_g \cup R^+) \times S^E$ is a discrete or a continuous transitions, such that:

The discrete transition $(m_D, m_C, t, v) \xrightarrow{a}_E (m'_D, m'_C, t', v')$ with $a \in \Sigma_g$ is subdivided into two forms:

- $a \in T^D$ (the firing of a transition T_j in the T-time PN) is the first form of discrete transitions: t_j is the time associated to the transition T_j ; t_k is the new time associated to the transitions newly enabled (T_k) after firing of T_j .

$(m_D, m_C, t, v) \xrightarrow{a}_E (m'_D, m'_C, t', v')$ if:

$$\left\{ \begin{array}{l} M \geq (., T_j), T_j \in T^{D2} \text{ and } m'_D = m_D - \text{Pre}(., T_j) + \text{Post}(., T_j), m'_C = m_C \\ \alpha(T_j) \leq t_j \leq \beta(T_j) \\ \text{if } \uparrow \text{ enabled}(T_k, M, T_j) : t'_k = 0, \text{ else } t'_k = t_k \\ \forall (P_k \in T_i^\circ), T_j \in T^{D2} \{T_i, T_{i+1}\} \in T^C, \text{ and } v'_i = 0 \\ \forall (P_j \in T_j^\circ), T_j \in T^{D1}, \text{ and } v'_{i+1} = V_{i+1} \end{array} \right. \quad 2$$

- $a \in \{m_i = 0\}$ (the emptying of a continuous place P_i) is the he second form of discrete transitions:

$(m_D, m_C, t, v) \xrightarrow{a}_E (m'_D, m'_C, t', v')$ if:

$$\left\{ \begin{array}{l} v' = f(m_C, t) \quad 3 \\ m_i = 0 \end{array} \right.$$

²To calculate instantaneous firing speeds see (R.David & Alla, 2010).

³The instantaneous firing speeds are calculated according to an algorithm presented in details in (R.David & Alla, 2010).

- The continuous transition is the time elapsing:

$$(m_D, m_C, t, v) \xrightarrow{d}_E (m'_D, m'_C, t', v') \text{ if:}$$

$$\left\{ \begin{array}{l} m'_C = m_C + B.d \\ t' = t + d \\ \forall T_k \in T^D, \text{ if } m \geq \text{Pre}(\cdot, T_k) \text{ then } t'_k \leq \beta(T_k) \\ \forall P_i \in P^C, \text{ if } m_i > 0 \text{ then } m'_i > 0 \end{array} \right.$$

After construction of timed transition systems of HA and EHPN, the bisimilarity between semantics of the two models will be demonstrated subsequently.

3.3. Timed similarity and bisimilarity

In this sub-section, the proof of the correctness of the translation algorithm is given. This will guaranty that the translation is correct and terminates. It will be proved that the semantics of an EHPN and its translated HA are timed bisimilar. The timed similarity and bisimilarity relations between both semantics are given bellow to introduce the proof.

Definition 7: Timed Similarity and Bisimilarity relation between EHPN and HA.

- Let $S_{EHPN} = (S^E, S_0^E, \Sigma_g, \longrightarrow_E)$ be the TTS of EHPN and $S_{HA} = (S^A, S_0^A, E, \longrightarrow_A)$ be the TTS of HA. Let Φ be a binary relation on $S^E \times S^A$. We write $s\Phi s'$ for $(s, s') \in \Phi$. Φ is a timed relation from S_{EHPN} by S_{HA} if the following assertions are checked:

$$\left\{ \begin{array}{l} \text{if } s_1 \in S_0^E, \text{ then } \exists s_2 \in S_0^A \text{ such that } s_1\Phi s_2; \\ \text{if } s_1 \xrightarrow{\delta}_E s'_1, \text{ with } \delta \in R^+ \text{ and } s_1\Phi s_2 \text{ then } \exists s_2 \xrightarrow{\delta}_A s'_2 \text{ such that } s'_1\Phi s'_2; \\ \text{if } s_1 \xrightarrow{a}_E s'_1, \text{ with } a \in \Sigma_g \text{ and } s_1\Phi s_2 \text{ then } \exists s_2 \xrightarrow{a}_A s'_2 \text{ with } a \in E \text{ such that } s'_1\Phi s'_2. \end{array} \right.$$
- $\forall (S_{EHPN}\Phi S_{HA}) \wedge (S_{HA}\Phi^{-1}S_{EHPN})$, then $S_{EHPN} \approx S_{HA}$. \approx is a bisimilarity relation between S_{EHPN} and S_{HA} .
- $\forall S_1 = (q_1, x_1) \in S^E$ is a state of S_{EHPN} , $S_2 = (q_2, x_2) \in S^A$ is a state of S_{HA} and $S_1\Phi S_2 \Leftrightarrow m_D = \nabla(q_2)$ and $x_1 = x_2$, where ∇ is a function associating to each discrete marking a given location.

Remark 3: Relation Φ expresses the time evolution given by the two identical dynamics of the EHPN and the HA. They are calculated from the firing conditions of both models: balance marking for the continuous evolution and time interval for the discrete one.

Theorem 1: The TTS of EHPN (S_{EHPN}) and The TTS of HA (S_{HA}) are timed bisimilar,

$$S_{EHPN} \approx S_{HA}. (\approx \text{ is a timed bisimilarity relation}).$$

Proof. $\forall S_1 = (q_1, x_1) \in S^E$ and $S_2 = (q_2, x_2) \in S^A$ such that $S_1\Phi S_2$, We write $S_1\Phi S'_1$ for $(S_1, S'_1) \in \Phi$. $S_{EHPN}\Phi S_{HA}$ if the following assertions are checked :

$$\left\{ \begin{array}{l} \text{if } S_1 \xrightarrow{d \in R^+}_E S'_1, \text{ then } \exists S_2 \xrightarrow{d \in R^+}_A S'_2 \text{ such that } S'_1\Phi S'_2; \\ \text{if } S_1 \xrightarrow{a \in \Sigma_g}_E S'_1, \text{ then } \exists S_2 \xrightarrow{a \in \Sigma_g}_A S'_2 \text{ such that } S'_1\Phi S'_2; \end{array} \right.$$

Continuous transitions:

According to the semantics of EHPN (Definition 6) the continuous transition $\xrightarrow{d \in R^+}_E$ is the time elapsing. t' and m'_C check the following: $t' = t + d$; $m'_C = m_C + B.d$ and $m'_D = m_D \Rightarrow q'_1 = q_1, x'_1 = F(d)$.

According to our algorithm the transition $\xrightarrow{d \in R^+}_E$ is translated to a transition $d = [\alpha_j \beta_j]$ between sub-location (Algorithm1, line 9) or between location (Algorithm1, line 29), so only dynamics that will be changed during this time, then

$$\begin{aligned}
x'_2 = F(d) \text{ and } \forall m'_D = m_D &\Leftrightarrow \nabla(q'_2) = \nabla(q_2) \Rightarrow q'_2 = q_2. \\
S_1 \Phi S_2, (q'_1 = q_1) \wedge (q'_2 = q_2) &\Rightarrow q'_1 = q_2 \\
x'_1 = F(d) \text{ and } x'_2 = F(d) &\Rightarrow x'_1 = x'_2
\end{aligned}$$

$$\Rightarrow S'_1 \Phi S'_2 \quad (1)$$

Discrete transitions: According to the semantics of EHPN, the discrete transition $\xrightarrow{a \in \Sigma_g}_E$ is:

- a) The firing of a transition T_j in the T-time PN is the discrete transition $\xrightarrow{a \in T^D}_E$:
if $M \geq \text{Pre}(\cdot, T_j) \Rightarrow M = g(x_1) = \text{true}$, and if T_j is fired, then $m'_C = m_C$; $m'_D = m_D - \text{Pre}(\cdot, T_j) + \text{Post}(\cdot, T_j) \Rightarrow \text{inv}(q'_2)(x'_1) = \text{true}$ and $\text{inv}(q_2)(x_1) = \text{true}$

According to our algorithm (Algorithm 1, line 5-9), T_j correspond to the transition with guard $m_i = \mathbf{S} \Rightarrow m_i = g(x_2)$

if $g(x_2) = \text{true}$ and $\text{inv}(q_2)(x_2) = \text{true}$, then $\text{inv}(q'_2)(x'_2) = \text{true}$.

$\text{inv}(q'_2)(x'_2) = \text{true}$ and $g(q'_2)(x'_2) = \text{true}$ such that $S_1 \Phi S_2$ then $x'_1 = x'_2$

$$\Rightarrow S'_1 \Phi S'_2 \quad (2)$$

- b) the emptying of a continuous place P_i is the discrete transition $\xrightarrow{a \in \{m_i=0\}}_E$: It is considered as a discrete event of type (C1-event), If $m_i = 0, g(x_1) = 0 \Rightarrow m_i = g(x_1) = \text{true} \Rightarrow m'_D = m_D$, and $m'_C = m_C - \text{Pre}(\cdot, P_i)$ then $g(q'_2)(x'_2) = \text{true}$ this emptying is translated in this work (Algorithm1, line 38) to a transition with a guard $m_i = 0 \Rightarrow g(x_2) = 0 \Rightarrow m_i = g(x_2)$.
If $g(x_2) = \text{true}$ and $\text{inv}(q_2)(x_2) = \text{true}$, then $g(q'_2)(x'_2) = \text{true}$, such that $S_1 \Phi S_2$ then $x'_1 = x'_2$

$$\Rightarrow S'_1 \Phi S'_2 \quad (3)$$

From (1), (2) and (3) we write: for all S_1 and S_2 on relation,

$$S_{EHPN} \Phi S_{HA} \quad (4)$$

With the same method, according to the semantics of HA (Definition 5) and according to our algorithm, it has been proved that:

$$S_{HA} \Phi^{-1} S_{EHPN} \quad (5)$$

From (4) and (5), it has been deduced that S_{EHPN} and S_{HA} are timed bisimilar for all S_1 and S_2 on relation. We write:

$$S_{EHPN} \approx S_{HA} \quad (6)$$

□

From (6) it has been concluded that the semantics of EHPN are timed bisimilar to semantics of HA, this means that a behavioral equivalence between EHPN and HA has been provided. This last ensures the correctness of our translation procedure presented in this work.

4. Case study

The liquid heating system (Derbel, 2009), illustrated in Figure 6(a), is composed of two valves V_1 and V_2 , a tank, a resistor, a thermostat and two level sensors: c_1 and c_2 monitor respectively the

maximal and the minimal levels. The liquid to be heated is introduced through the valve V_1 with a flow rate 5 v.u./t.u. (volume units per time unit). When the liquid level reaches the maximum level 500 v.u., the sensor c_1 generates a notification event to controller for the closing of valve V_1 . Thereafter, the liquid is heated for a period of 40t.u. Then the liquid is discharged through the valve V_2 with a flow rate 8 v.u./t.u. until the tank is empty. At this time, the sensor c_2 generates a notification event; therefore the controller starts a new heating cycle. Assuming that the functioning of this system can be affected by two faults:

- The first fault corresponds to existence of a leak in the tank with a flow rate 0.5 v.u./t.u. This fault is represented by the unobservable event σ_1 ;
- The second fault corresponds to the blocking valve V_1 in the closed position. This fault is represented by the unobservable event σ_2 .

s_1 and s_2 are observable events and they correspond to notifications generated, respectively, by the sensors c_1 and c_2 . The event u is introduced to allow the representation of an autonomous transition. At the initial time $t = 0$, the tank is empty.

The elementary HPN model in normal functioning is given in Figure 6(b). With the transitions T_4 and T_5 are associated maximal firing speeds $V_4 = 5$ and $V_5 = 8$; and with the transitions T_1 , T_2 and T_3 are associated durations $d_1 = 0$, $d_2 = 40$ and $d_3 = 0$ of events s_1 , u and s_2 successively, knowing that d_1 and d_3 are a synchronization durations only. The occurring of leak fault of continuous aspect is modelled by the continuous transition T_2 attached to the continuous place P_3 with an arc from the place to the transition T_2 . This transition is controlled by the occurrence of the unobservable event σ_1 associated to the discrete transition T_1 (Figure 6(c)).

The blocking valve fault is modelled by the discrete transition T_1 attached to the discrete place P_2 with an arc from the place to the transition T_1 . This transition is controlled by the occurrence of the unobservable event σ_2 associated to the discrete transition T_1 , this fault can occurred at any time during filling process (Figure 6(d)). To get the global model given in Figure 7, it is necessary to perform a structural composition of normal functioning model and fault models. The model with faults is translated directly using the presented algorithm, the final HA is obtained as follows:

- Step 1: Build the reachable marking graph of subjacent autonomous PN (Figure 8) by isolating the time PN part from continuous part (CCPN) of hybrid model;
- Step 2: Associate respectively the dynamics and the guards to each location and transition (Figure 10) according to the continuous part configuration, where each reachable discrete marking correspond to a CCPN configuration (Figure 9) and null flow vectors are suppressed;
- Step 3: The value of the entry guard determines the target location whose downstream configuration of each location, the unreachable locations must be deleted (this is not the case in our example). The final HA is given in Figure 11.

By applying this translation to the case study, a HA model bisimilar to EHPN model is obtained, then all the properties on the HA remain true for the EHPN. Moreover, the ease of faults modelling that affect the continuous part of heating system, and the change of system dynamics caused by these faults are shown.

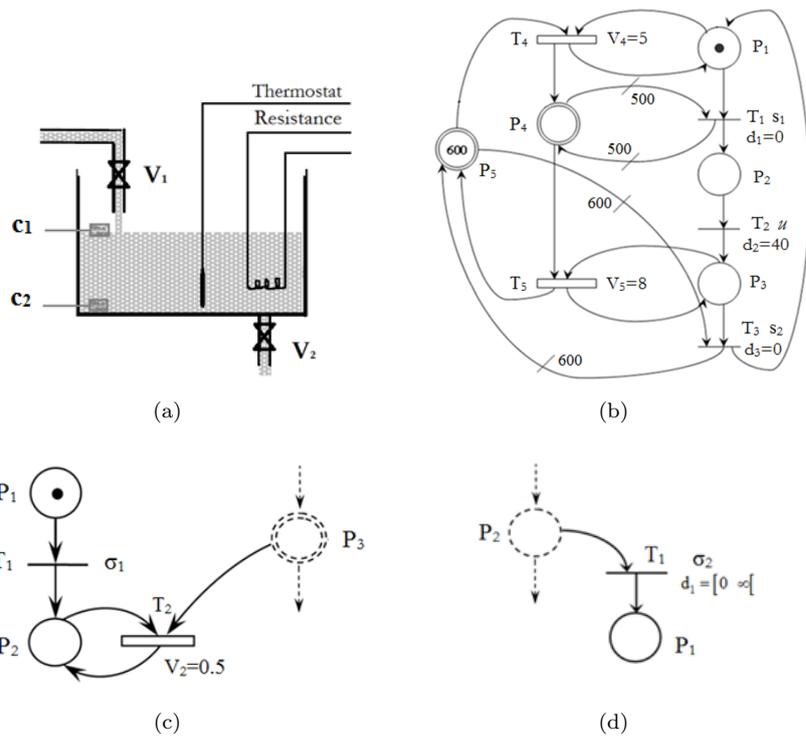


Figure 6. (a). Liquid heating system. (b) Elementary HPN model in normal functioning. (c) First fault model.(d) Second fault model.

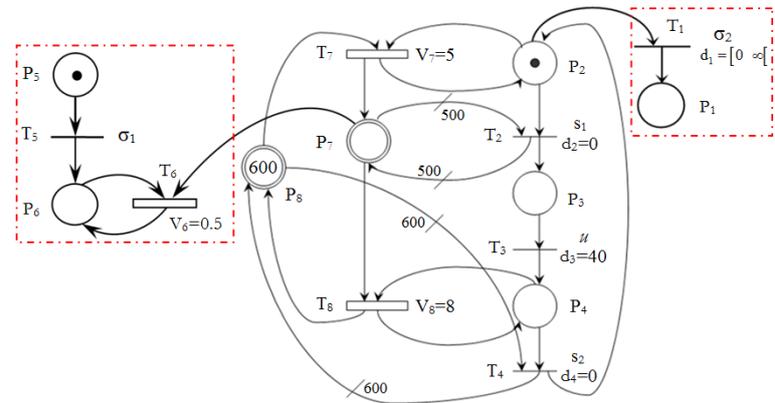


Figure 7. Global elementary HPN model.

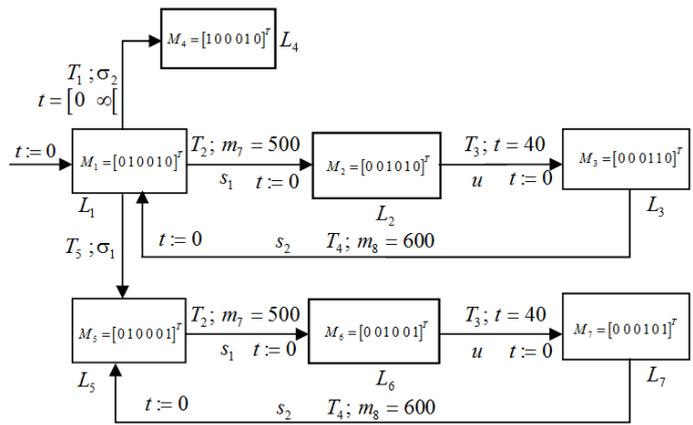


Figure 8. The reachable marking graph of subjacent autonomous PN.

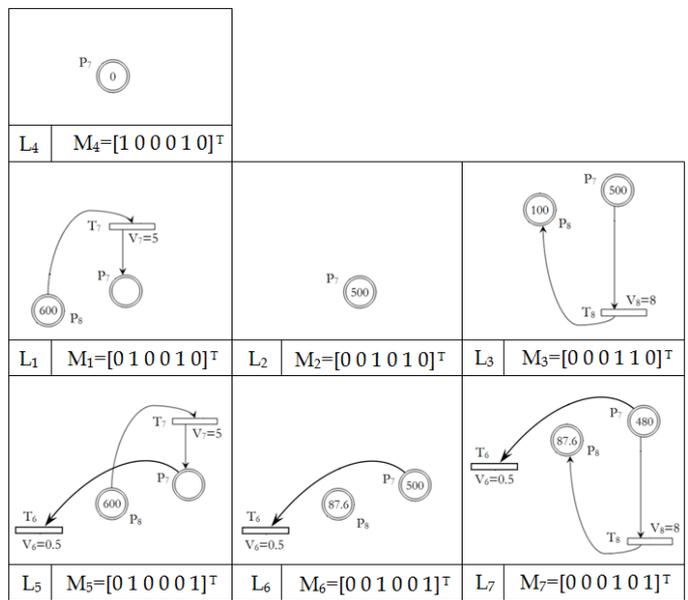


Figure 9. CCPN configuration of each reachable discrete marking.

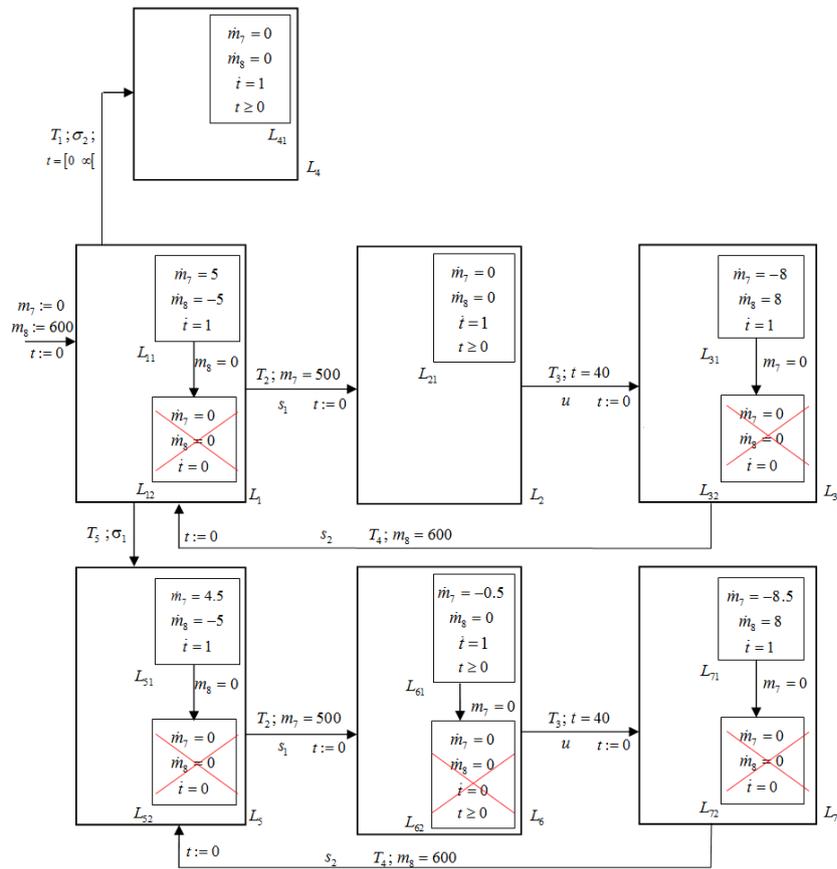


Figure 10. Initial hybrid automaton.

A structural modelling has been done and the same model presented in (Derbel, 2009) is obtained, with an automatic and systematic way without imagining faults behaviour. In addition, the dynamics gap appears clearly on locations of the faulty behaviour. This gap can be observed on locations \$L_5, L_6\$ and \$L_7\$ and it could be remarked on the dynamic value which becomes respectively 4.5 instead of 5, -0.5 instead of 0 and -8.5 instead of -8 (subtractive flow). This corresponds to the basic idea which will be used for the diagnoser synthesis.

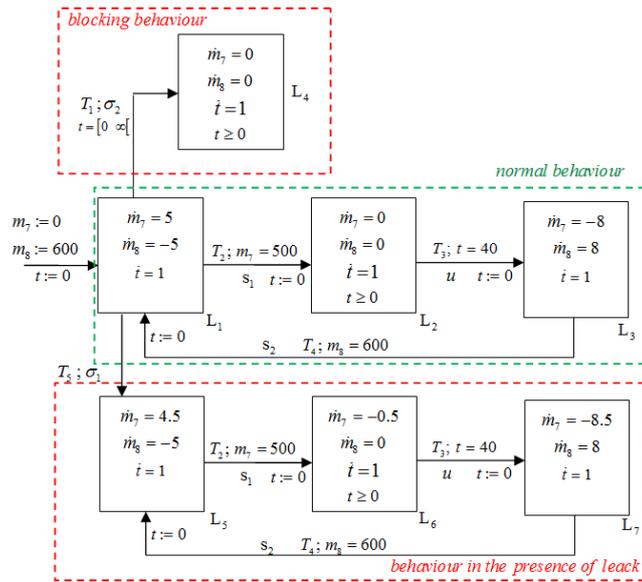


Figure 11. The final hybrid automaton of the liquid heating system.

5. Conclusion

In this paper, EHPNs is used to describe the evolution in normal operation of a type of HDSs called continuous flow systems. A new faults modelling method based on fault structure is proposed to model faults that can be occurred in this type of systems. This has allowed us to get the global model by performing a structural composition of both models, the physical system model in normal functioning with the faults model. No dynamics synchronization is used. Then, the model with faults is translated directly to get an automaton model ready for analysis using a new translation approach. Timed bisimilarity property between EHPN and HA as TTS models has been proved to ensure the correctness of our translation algorithm. A liquid heating system has illustrated the proposed approach. As future work, a diagnoser synthesis of this type of systems modelled by EHPNs could be developed.

Acknowledgements

We warmly thank Latéfa Ghomri Doctor at University of Tlemcen, Algeria, and Yamen El Touati Doctor at university of Manouba, Tunisia for their support and guidance in Section 3.3 of this paper.

References

Allam, M., & Alla, H. (1998). From hybrid petri nets to hybrid automata. *Journal European des Systemes Automatiss.*

Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T. A., Ho, P., Nicollin, X., . . . Yovine, S. (1995). The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1), 3-34.

Alur, R., Courcoubetis, C., Henzinger, T., & Ho, P. (1993). Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L.Grossman, A. Nerode, A.P.Ravn, and H. Rischel, editors, *Hybrid Systems I, Lecture Notes in Science*, 736, 209-229.

Alur, R., & Dill, D. L. (1994). A theory of timed automata. *Theor. Comput. Sci.*, 126(2), 183-235.

- Bail, J. L., Alla, H., & David, R. (1991). Hybrid petri nets. In *Proceedings of the european control conference grenoble* (pp. 1472–1477).
- Berthomieu, B., & Menasche, M. (1983).
In *Ifip congress series* (Vol. Vol.9, p. 41-46).
- Bornot, S., Sifakis, J., & Tripakis, S. (1998). Modeling urgency in timed systems. In W.-P. de Roever, H. Langmaack, & A. Pnueli (Eds.), *Compositionality: The significant difference* (Vol. 1536, p. 103-129). Springer Berlin Heidelberg.
- Branicky, M. S. (1995). *Studies in hybrid systems: Modeling, analysis and control* (Unpublished doctoral dissertation). MIT, MA.
- Buisson, J., & Cormerais, H. (1998). Descriptor systems for the knowledge modelling and simulation of hybrid physical systems. *Journal Européen des systèmes automatisés*, 32(9-10), 1047-1072.
- Cassez, F., & Roux, O. H. (2006). Structural translation from time petri nets to timed automata. *Journal of Systems and Software*, 79(10), 1456–1468.
- Cortes, L., Eles, P., & Peng, Z. (2000). Verification of embedded systems using a petri net based representation. In *System synthesis, 2000. proceedings. the 13th international symposium on* (p. 149-155).
- David, R., & Alla, H. (2001). On hybrid petri nets. *Discrete Event Dynamic Systems*, 11(1-2), 9-40.
- Demongodin, I. (2001). Generalised batches petri net: Hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1-2), 137–162.
- Demongodin, I., & Rouibia, S. (2003). Modélisation par réseaux de petri lots et analyse de l'état stable par automates hybrides. In *4e conférence francophone de modélisation et simulation mosim03*.
- Derbel, H. (2009). *Diagnostic à base de modèles des systèmes temporisés et d'une sous-classe de systèmes dynamiques hybride* (Unpublished doctoral dissertation). université joseph fourrier.
- Derbel, H., Alla, H., Ben Hadj-Alouane, N., & Yeddes, M. (2009, June). Online Diagnosis of Systems with Rectangular Hybrid Automata Models. In *13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM09)* (p. 123-129). Moscou, Russia.
- Etouati, Y., Yeddes, M., Hadj Alouane, N., & Alla, H. (2009, February). Du réseau de Petri temporel étendu vers les automates hybrides linéaires pour l'analyse des systèmes. In *IEEE Conférence Internationale Francophone d'Automatique, CIFA 2009* (p. 00). Bucarest, Romania.
- Ghomri, L. (2012). *Synthèse de contrôleur de systèmes hybrides à flux continu par réseaux de petri hybrides* (Unpublished doctoral dissertation). universite of Abou-Bekr BelkadTlemcen, Algeria.
- Jeron, T., Marchand, H., Pinchinat, S., & Cordier, M.-O. (2006, July). Supervision patterns in discrete event systems diagnosis. In *Discrete event systems, 2006 8th international workshop on* (p. 262-268).
- Larsen, K. G., Petterson, P., & Yi, W. (1995). Model-checking for real-time systems. In *Proceedings of the 10th international symposium on fundamentals of computation theory* (pp. 62–88). London, UK, UK: Springer-Verlag.
- Lime, D., & Roux, O. H. (2003, September). State class timed automaton of a time Petri net. In *The 10th international workshop on Petri nets and performance models, (pnpm'03)* (pp. 124–133). Urbana, USA: IEEE Computer Society press. ((See extended version in *Journal of Discrete Event Dynamic Systems*))
- Liu, H.-C., Lin, Q.-L., & Ren, M.-L. (2013, December). Fault diagnosis and cause analysis using fuzzy evidential reasoning approach and dynamic adaptive fuzzy petri nets. *Comput. Ind. Eng.*, 66(4), 899–908.
- Pandalai, D., & Holloway, L. (2000, May). Template languages for fault monitoring of timed discrete event processes. *Automatic Control, IEEE Transactions on*, 45(5), 868-882.
- Petterson, S., Lennartson, B., et al. (1995). Hybrid modelling focused on hybrid petri nets. *Proceedings 2nd European Workshop on Real-time and Hybrid systems*, 303–309.
- Ramchandani, C. (1974). *Analysis of asynchronous concurrent systems by timed petri nets* (Unpublished doctoral dissertation). Massachusetts Institute of Technology, Cambridge, MA.
- R.David, & Alla, H. (2010). *Discrete, continuous, and hybrid petri nets*. Springer.
- Renganathan, K., & Bhaskar, V. (2011). An observer based approach for achieving fault diagnosis and fault tolerant control of systems modeled as hybrid petri nets. *{ISA} Transactions*, 50(3), 443 - 453.
- Renganathan, K., & Bhaskar, V. (2013). Modeling, analysis and performance evaluation for fault diagnosis and fault tolerant control in bottle-filling plant modeled using hybrid petri nets. *Applied Mathematical Modelling*, 37(7), 4842–4859.
- Roth, M., Lesage, J.-J., & Litz, L. (2011). The concept of residuals for fault localization in discrete event

- systems. *Control Engineering Practice*, 19(9), 978 - 988. (Special Section: DCDS09 The 2nd {IFAC} Workshop on Dependable Control of Discrete Systems)
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995, Sep). Diagnosability of discrete-event systems. *Automatic Control, IEEE Transactions on*, 40(9), 1555-1575.
- Sava, A. (2001). *Sur la synthèse de la commande des systèmes à événements discrets temporisés* (Unpublished doctoral dissertation). INPG, Grenoble, France.
- Sayed-Mouchaweh, M. (2012). Decentralized fault free model approach for fault detection and isolation of discrete event systems. *European Journal of Control*, 18(1), 82-93.
- Sayed Mouchaweh, M., Philippot, A., & Carré-Ménétrier, V. (2008). Decentralized diagnosis by boolean discrete event system model: Application on manufacturing systems. *International Journal of Production Research (IJPR), Special Issue*, 46(19), 1555-1575.
- Sifakis, J., & Yovine, S. (1996). Compositional specification of timed systems. In *Stacs 96* (pp. 345–359). Springer.
- Zad, S. H., Kwong, R., & Wonham, W. (1999). Fault diagnosis in finite-state automata and timed discrete-event systems. In *In 38th ieee conference on decision and control*.

Appendix 1: Translation algorithm.

The translation algorithm of an EHPN into a HA of this work presented in Section 3.1 is detailed on the following algorithm:

Algorithm 1 Translation algorithm

« Initialization »;

Isolate the timed PN part from continuous part (CCPN) of hybrid model;

Build the reachable marking graph of Subjacent autonomous PN;

for $\forall P_i \in P^C$, $\text{card}(T^{D2} \cap P_i^o) \leq 1$ **do**

if the place $P_i \in P^D$ source of transition $T_j \in T^{D2}$ has no output transition $T_j \in T^{D1}$ **then**

Burst each location which corresponds to $T_j \in T^{D2}$ into two sub-locations L_{k1} and L_{k2} ;

Associate the time clock to location L_{k2} with initialization at his input;

Connect L_{k1} and L_{k2} for each location with a transition and associate him the guard $m_i = \mathbf{S}$;

Create an output transition for each L_{k2} and associate him the guard $d_j = [\alpha_j \beta_j]$;

if $d_j = [0 \ 0]$ corresponds to $T_j \in T^{D2}$ **then**

Eliminate the location L_{k1} for each location burst;

Connect each location L_{k2} to the next location, by keeping the guard $m_i = \mathbf{S}$;

else

Connect the output transition of each L_{k2} with the next location or sub-location;

end if

else

Burst each location which corresponds to $T_j \in T^{D2}$ into two sub-locations L_{k1} and L_{k2} ;

Associate the time clock to location L_{k2} with initialization at his input;

Connect L_{k1} and L_{k2} for each location with a transition and associate him the guard $m_i = \mathbf{S}$;

Create an output transition for each L_{k2} and associate him the guard $d_j = [\alpha_j \beta_j]$;

```

if  $d_j = [0 \ 0]$  corresponds to  $T_j \in T^{D2}$  then
  Eliminate the location  $L_{k1}$  for each location burst;
  Connect each location  $L_{k2}$  to the next location, by keeping the guard  $m_i = S$ ;
  Create for each output transition  $T_j \in T^{D1}$  a destination location;
  Connect each destination location created to the location  $L_{k2}$ , with a transition and
  associate him the guard  $d_j = [\alpha_j \ \beta_j]$ ;
else
  Connect the output transition of each  $L_{k2}$  with the next location;
  Create for each output transition  $T_j \in T^{D1}$  a destination location;
  Connect each destination location created with two transitions of  $L_{k1}$  and  $L_{k2}$ , and asso-
  ciate them the guard  $d_j = [\alpha_j \ \beta_j]$ ;
end if
end if
end for
« CCPN_configuration », configure the continuous part of CCPN for each discrete marking;
put  $m_0^* = P^C$ ;
create  $L_{f0}$ , for each location and associate him the activity  $\dot{M} = W.V$ ;
« marking_verification ».
if the marking for each C-place  $P_i$  is decreasing then
  Create from  $L_{f0}$ , a transition for each place  $P_i$  and affect him the guard  $m_i = 0$ ;
  For each transition previously built, create a destination location and associate him the activity
   $\dot{M} = W.v(t)$ ;
  Merge locations having the same activity;
  if  $\dot{m}_i = 0$ , for each  $P_i$  whose marking is initially decreasing then
    Go to « comparison_vectors »;
  else
    Return to « marking_verification »;
  end if
end if « comparison_vectors », Replace the transitions by comparing the Boolean vectors;
Calculate the Boolean vector  $\bar{M}$  for each internal location;
if  $\bar{M}_l \geq \bar{M}_{l+1}$  then
  Replace each transition between locations by a transition between their internal locations
   $L_l, L_{l+1}$ ;
end if
End

```
