SYMBOLIC TEMPORAL CONSTRAINT ANALYSIS, AN APPROACH FOR VERIFYING HYBRID SYSTEMS

Nicolas Rivière * **Hamid Demmou** * **Robert Valette** *
Malika Medjoudj *

* *LAAS-CNRS, F-31077 Toulouse Cedex 4, France*

Abstract: The purpose of the paper is to illustrate a method, based on theorem proving, allowing the determination of a set of constraints such that some property of an hybrid system is verified. The approach is based on the generation of scenarios by proving some linear logic sequents and on the analysis of symbolic temporal constraints in a Simple Temporal Network. In the presented example, the property is the reachability of a given state within some temporal constraint.

Keywords: Hybrid system, Verification, Constraint Satisfaction Problems, Petri nets

1. INTRODUCTION

The objective of this paper is to illustrate, on a non trivial example, an approach for proving some properties of hybrid systems. The originality resides in the following points. The technique belongs to theorem proving and not to property verification. It is based on an exploration of the trajectory space (scenarios leading to some specified states) and not on state space exploration. Finally, in place of verifying a property, the approach provides a set of constraints on some parameters allowing the proof of the property. The property, considered in this paper, is that the lapse of time to reach a given state belongs to a specified domain. Only the principles of the method are given, all the technical details can be found in (Rivière, 2003). This paper focuses on the presentation of the example.

2. PRINCIPLES OF THE APPROACH

2.1 General view

The hybrid system is modeled by means of a Petri net, for its discrete view and by means of sets of differential equations (linear if possible) or temporal abstractions for the continuous dynamics. Then, a p-invariant based analysis is done to establish relations between token locations and continuous variable domains. After this step, the scenarios (consistent with the discrete view) corresponding to the property are exhaustively generated. This step is based on a translation of the reachability problem into the proof of a linear logic sequent. The next step consists in deriving, for each *scenario* - a set of transition firings with a partial order - a Simple Temporal Network (STP) by taking into account the quantitative temporal constraints expressed by means of labels attached to the Petri net (Mancel *et al.*, 2002). Finally, by a constraint propagation of the symbolic temporal constraints, a symbolic domain is derived for the date of the firing of the last transition of the scenario. The upper and lower bound of the domain involve parameters of the problem and by matching this domain with the required one, the set of constraints which have to be verified by the parameters is derived. Let us detail two critical aspects: the translation of the Petri net into linear logic and the generation of a scenario by means of a proof.

2.2 Translation into linear logic

The Multiplicative Intuitionist fragment of Linear logic (MILL) (Girard, 1987) is sufficient. It only contains the multiplicative connective "$\otimes$" (conjunction of hypotheses) and the linear implication "$\multimap$". There

is no negation and the meta connective "," is commutative. The specificity of Linear logic (with respect to classical logic) is that logical propositions are consumed when they are used for a deduction. Proving a sequent is verifying that the required hypotheses are available when they are used in a proof step.

Atoms denote tokens and their names are those of the places where they are located. Markings are denoted by monomial in $\otimes$. *Transition firings* are denoted by formulas of the form: $t : Pre(t) \multimap Post(t)$ where $Pre(t)$ and $Post(t)$ are monomials in $\otimes$ (in the same way as markings). A reachability proof is expressed by the following sequent: $M_0, \lambda_0 \vdash M_n$ where M_0 is the initial marking, M_n is the final marking and λ_0 is a set of formulas denoting *transition firings* specified as above. Each formula denotes a transition firing. If a transition t is fired n times then the corresponding formula has to be present in n exemplars in λ_0.

2.3 Proof of a sequent and labeling the proof tree

A reachability proof is equivalent to the proof of the corresponding sequent. Its purpose is to derive the partial order on the set λ_0 defining the reachability scenario. A sequent proof tree is a syntactical proof. It is a set of rules proving that the connectives have been correctly introduced. The canonical construction of the proof tree is based on an iterative step which consists in eliminating each transition firing of λ_0 after having verified that the required atoms have been produced. This step has to be executed once for each transition firing of the list and there is a bijection between it and each transition firing in λ_0. The precedence relations imposed by the structure and the markings of the Petri net are those which relate, *for each atom*, the application of the iterative step which produces it with the one which consumes it (Mancel *et al.*, 2002). These precedence relations are obtained by *labeling the proof tree* (Rivière, 2003). They formally define one scenario.

3. EXAMPLE

The example is a modified version of a benchmark proposed by the French working group STRQDS (STRQDS, 2002), which has been first presented by (Boniol and Carcenac, 2002) and first studied by (Villani, 2004) in its hybrid version. The purpose is to verify a landing system for an airplane. It is composed of three landing gears which have to be extended for landing and retracted for flying fast. Each gear is in a box closed by a door. Before extending or retracting a gear the box door has to be opened and it is automatically closed when the movement is complete.

The gears are controlled by means of a three-position command. When the command is E, the box doors are opened, the three gears are extended and the doors are

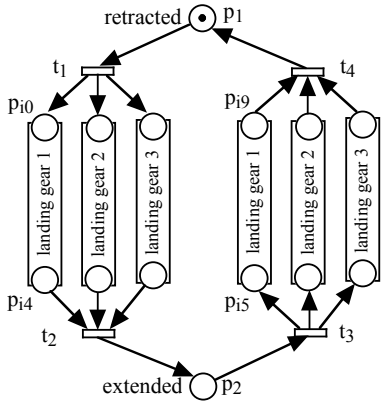

Fig. 1. General view of the net

closed. When the command is R, the box doors are opened, the three gears are retracted and the doors are closed. In the intermediary position the command is B and the gears are blocked in their current positions. We assume that opening and closing movements of the doors go until completion in any case.

The property to be verified is that when the pilot maintains the E command the lapse of time required to reach the final state (ready for landing) is bounded by Δ whatever the commands which have been previously done (and therefore whatever the current state of the landing system).

3.1 Global Petri net model

A global view of the model is given in figure1. Place p_1 represents the state in which the three gears are retracted. Transition t_1 is fired when the command E is issued in this state. This event generates three concurrent branches, one for each gear. Transition t_2 is fired when the three gears are extended. The right part of the figure correspond to the retracting command.

3.2 Petri net model for extending

A detailed view of an extending gear behavior (gear i) is represented in figure 2. Note that places p_{i0} and p_{i4} are the same as in figure 1. In a preceding paper (Boniol and Carcenac, 2002) the system has been represented by means of Lustre, Esterel and timed automata. In this paper it has been chosen to take into account the hybrid nature of the system by means of a model based on Petri nets and differential equations. It has also been assumed that the gears could be blocked in any position. However we have not chosen the same level of detail as in (Villani, 2004) in order to be able to give a simple proof. In particular, the three gears have been assumed to be independent.

We assume that the gear continuous dynamics could be approximated with a delay and two linear behaviors as represented in figure 3. The delay is delimited by a

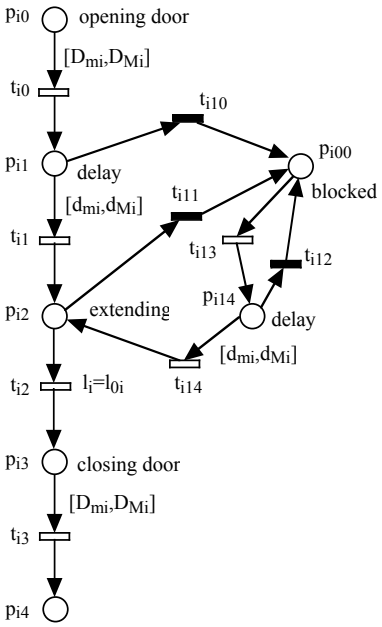

Fig. 2. Petri net for extending gear i

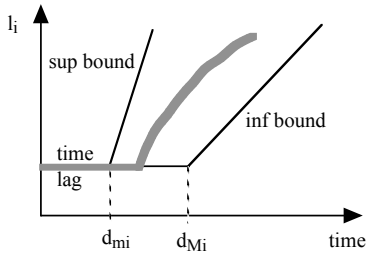

Fig. 3. Gear continuous dynamics

time interval $[d_{mi}, d_{Mi}]$ which means that it may vary between the minimal value d_{mi} and the maximal one d_{Mi}. The slope of the lower envelope is a_i and that of the upper one is b_i.

Place p_{i0} corresponds to the opening of the box door. Its continuous dynamics is taken into account by means of a temporal abstraction. The minimal duration for opening or closing the box of gear i is D_{mi} and its maximal one is D_{Mi}. The corresponding time constraint is represented by the interval $[D_{mi}, D_{Mi}]$ attached to the arc (p_{i0}, t_{i0}). This means that the token in $(p_{i0}$ has to remain in this place at least D_{mi} and at most D_{Mi}. As there are no other transition which can consume it, the token cannot remain in p_{i0} after D_{Mi}. This model is a pt-arc-time Petri net with respect to time (timing constraints are attached to the arcs linking places to transitions). It is very similar to p-time Petri nets in which temporal constraints are attached to places. The unique difference is that the temporal constraint associated with the place may differ with respect to each output transition of the place.

Place p_{i1} represents the time lag for the movement of gear i. The corresponding time constraint is represented by the interval $[d_{mi}, d_{Mi}]$ attached to the arc (p_{i1}, t_{i1}).

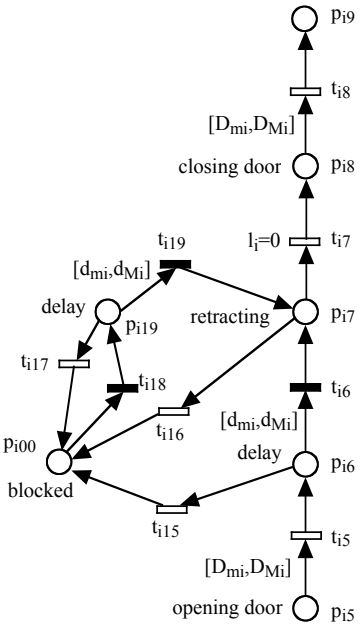

Fig. 4. Petri net for retracting gear i

Place p_{i2} represents the linear interpolation of the gear movement. The differential equation

$$dl_i/dt = c_i \text{ with } c_i \in [a_i, b_i] \text{ and } a_i \geq 0 \qquad (1)$$

is attached to this place. The Petri net is therefore a predicate differential one as defined in (Champagnat *et al.*, 2001) and used in (Villani *et al.*, 2003). The current value of each gear position l_i is attached to the corresponding token. Transition t_{i2} is fired when $l_i = l_{0i}$. It is *exactly* fired at the time corresponding to this condition.

Finally, place p_{i3} corresponds to the door being closed (same temporal constraint as for p_{i0}) and place p_{i4} to the necessary wait state before the synchronization between the three gears represented by t_2.

The right part of the net represents the states reached when the command is not held in E position by the pilot. When the pilot commutes to R or B the activities represented by places p_{i1} and p_{i2} are *immediately* interrupted (transitions t_{i10} or t_{i11}). The token is put in place p_{i00} and the current value of l_i is stored. If command E is emitted in this state, the extension process is resumed by means of transitions t_{i13} and t_{i14}. A new initial delay is imposed by place p_{i14}, delay which can be interrupted by the firing of t_{i12} if the pilot moves the command to B or R again.

3.3 Petri net model for retracting

The model is similar and is represented in figure 4. An important point is that place p_{i00} is shared between the two Petri nets in figures 2 and 4. The blocked state is the same because the pilot may suddenly pass from a retracting to an extending operation and vice-versa. Places p_{i5} and p_{i9} are the same as in figure 1.

The differential equation attached to place p_{i7} is:

$$dl_i/dt = -c_i \text{ with } c_i \in [a_i, b_i] \text{ and } a_i \geq 0 \quad (2)$$

It is important to point out that the threshold with which the firing of transition t_{i7} is synchronized is $l_i = 0$.

3.4 Invariant analysis

By replacing the rectangles *"landing gear i"* in figure 1 by the subnets in figures 2 and 4 it can easily be proven that there are three positive p-invariants: for each landing gear, the sum of the markings of all the places increased by the markings of places p_1 and p_2 is equal to 1. This means that the Petri net is 1-bounded.

Only six places have differential equations attached to them: the two places p_{i2} and p_{i7} for each gear. This means that the values of variables l_i representing the gear positions only vary when one of these places contains a token. Given one gear (given a value of i), l_i cannot simultaneously be increased and decreased because it is not possible to simultaneously have a token in place p_{i2} and a token in place p_{i7} (straightforward consequence of the above p-invariants). It is assumed that the initial value of these three continuous variables (when $M(p_1) = 1$) is 0.

3.5 Delimiting continuous variable domains

By a backward reasoning starting from place p_1, it can be proven that any scenario allowing the production of a token in place p_1 terminates by the unique scenario s_1 characterized by the sequent:

$$p_{17} \otimes p_{27} \otimes p_{37}, t_{17}, t_{27}, t_{37}, t_{18}, t_{28}, t_{38}, t_4 \vdash p_1 \quad (3)$$

As transitions t_{i7} are only fired when $l_i = 0$, and as in places p_{i8}, p_{i9} and p_1 variables l_i are constant, then necessarily $\forall i, l_i = 0$ when $M(p_1) = 1$. Similarly, if $M(p_2) = 1$ then $\forall i, l_i = l_{0i}$. The position of the gear i is also $l_i = l_{0i}$ when there is a token in the places p_{i5} or p_{i6} and $l_i = 0$ when there is a token in the places p_{i0} or p_{i1}.

Let us now consider the following hypothesis:

$$\forall M \; reachable \; marking \; , \forall i, 0 \leq l_i \leq l_{0i} \quad (4)$$

The proof is straightforward. Variable l_i only increases when $M(p_{i2}) = 1$. During this *activity* transition t_{i2} remains enabled and is fired as soon as the value $l_i = l_{0i}$ is reached. Instantaneously the token in place p_{i2} is removed and the value of l_i remains constant. For the minimal bound of l_i a similar reasoning involving place p_{i7} and transition t_{i7} can be done. As condition 4 is initially true, it is true for any reachable state.

3.6 Building elementary scenarios

In order to prove the property, it is necessary to derive all the scenarios leading to the state characterized by the marking $M(p_2) = 1$ (which entails $\forall i, l_i = l_{0i}$ and therefore correspond to a unique state). Only events consistent with the command E are taken into account. The scenarios are built in a *modular* way in order to cope with the search space explosion and to exploit *system symmetries*.

Any scenario allowing the production of a token in place p_2 terminates by the unique scenario s_2 characterized by the sequent:

$$s_2 \; : \; p_{14} \otimes p_{24} \otimes p_{34}, t_2 \vdash p_2 \quad (5)$$

In order to exploit symmetry, the three similar scenarios leading to one token in places p_{i4} are independently built. In a first step, the scenarios are derived from the Petri net in figure 2 by considering that the transitions t_{i10}, t_{i11} and t_{i12} cannot be fired (inconsistency with the command E). The following scenarios are derived:

$$s_3 \; : \; p_{i0}, t_{i0}, t_{i1}, t_{i2}, t_{i3} \vdash p_{i4} \quad (6)$$

$$s_4 \; : \; p_{i00}, t_{i13}, t_{i14}, t_{i2}, t_{i3} \vdash p_{i4} \quad (7)$$

Place p_{i00} is shared between the nets in figures 2 and 4. This implies that scenario s_4 has to be extended by all the scenarios in figure 4 which produce a token in p_{i00} (remember that transitions t_{i10}, t_{i11} and t_{i12} cannot be fired). The transitions which cannot be fired in figure 4 are t_{i6}, t_{i18} and t_{i19}. The three following scenarios (s_5, s_6 and s_7 respectively) are derived:

$$s_5 \; : \; p_{i19}, t_{i17} \vdash p_{i00} \quad (8)$$

$$s_6 \; : \; p_{i7}, t_{i16} \vdash p_{i00} \quad (9)$$

$$s_7 \; : \; p_{i5}, t_{i5}, t_{i15} \vdash p_{i00} \quad (10)$$

Finally, as in the global net in figure 1 transition t_1 can be fired (with command E) but not transition t_3, we have also to consider the scenario s_8:

$$s_8 \; : \; p_1, t_1 \vdash p_{10} \otimes p_{20} \otimes p_{30} \quad (11)$$

3.7 Building elementary Simple Temporal Networks

In order to derive the Simple Temporal Networks (STN) (Dechter *et al.*, 1991) corresponding to the scenarios, it is necessary to replace the continuous dynamics attached to places p_{i2} and p_{i7} by their *temporal abstractions*. We point out the fact that, *in our approach*, the time interval are defined *in a symbolic way* and *not in a numeric way* in order to go back to the hybrid expression when necessary. As a consequence, the continuous dynamics is taken into account by attaching the temporal constraints $[d_{2mi}, d_{2Mi}]$ to the arc (p_{i2}, t_{i2}) and $[d_{7mi}, d_{7Mi}]$ to (p_{i7}, t_{i7}).

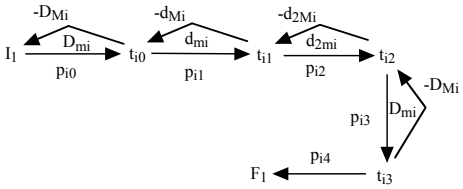

Fig. 5. STN associated with scenario s_3

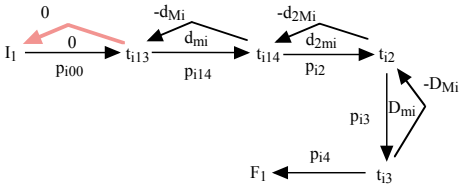

Fig. 6. STN associated with scenario s_4

Let us consider the scenario s_3 (sequent 6). The corresponding Simple Temporal Network is represented in figure 5. The initial node (event) I_1 represents the production of a token in place p_{i0}. This event coincides with the firing of t_1. The nodes denotes variables corresponding to the firing dates of the corresponding transitions, the arcs denotes the temporal constraints which have to be verified by these variables.

If arc labels such as d_{mi} or d_{2mi} are replaced by numerical values, classical Floyd-Warshall algorithm allows to derive the exact temporal constraint between two events ensuring 3-consistency. In the presented approach, the temporal constraint between the initial event I_1 and the one corresponding to the production of the last token *i.e* t_{i3} can be derived by means of a simple symbolic calculus. It is straightforward because there is only one oriented path in each direction between these two events. The obtained constraint is (x_{i3} is the variable denoting the date of event t_{i3} and x_1^3 the one of the initial event I_1 of scenario s_3):

$$2.D_{mi} + d_{mi} + d_{2mi} \leq x_{i3} - x_1^3 \leq 2.D_{Mi} + d_{Mi} + d_{2Mi} \quad (12)$$

The Simple Temporal Network corresponding to scenario s_4 (sequent 7) is given in figure 6. Event I_1 either corresponds to the production of a token in place p_{i00} or to the fact that a command E is executed when place p_{i00} contains a token. In the two cases the token cannot remain in place p_{i00} and this is expressed by adding an arc of length 0 between t_{i13} and I_1. The obtained constraint is (with the same notation as in 12):

$$D_{mi} + d_{mi} + d_{2mi} \leq x_{i3} - x_1^4 \leq D_{Mi} + d_{Mi} + d_{2Mi} \quad (13)$$

The Simple Temporal Network corresponding to scenario s_5 (sequent 8) is given in figure 7. Event I_1 corresponds to the execution of E when there is a token in place p_{i19}. For the same reason (command E has to be taken into account immediately) an arc

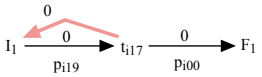

Fig. 7. STN associated with scenario s_5

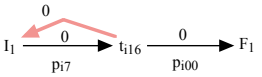

Fig. 8. STN associated with scenario s_6

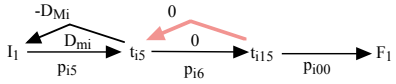

Fig. 9. STN associated with scenario s_7

of length 0 has been added between event t_{i17} and I_1. The following constraint is derived:

$$0 \leq x_{i17} - x_1^5 \leq 0 \quad (14)$$

The case of scenario s_6 (sequent 9) is similar. The Simple Temporal Network is represented in figure 8 and the following constraint is derived:

$$0 \leq x_{i16} - x_1^6 \leq 0 \quad (15)$$

Finally, the case of scenario s_7 (sequent 10) is also similar and the Simple Temporal Network is represented in figure 9. The derived constraint is

$$D_{mi} \leq x_{i15} - x_1^7 \leq D_{Mi} \quad (16)$$

3.8 Building significant scenarios

The significant event (producing the final state) is the firing of transition t_2. As t_2 has to be fired as soon as it is enabled, from scenario s_2 (5) it can be derived that:

$$x_2 = \max(x_{13}, x_{23}, x_{33}) \quad (17)$$

The sequential composition of the elementary scenarios s_5 (sequent 8 and STN 7) with s_4 (sequent 7 and STN 6) implies the fusion of the final event of the first scenario with the first event of the second scenario. This means that $x_{i17} = x_1^4$. In consequence:

$$D_{mi} + d_{mi} + d_{2mi} \leq x_{i3} - x_1^5 \leq D_{Mi} + d_{Mi} + d_{2Mi} \quad (18)$$

The sequential composition of s_6 (sequent 9 and STN 8) with s_4 (7 and STN 6) results in the same constraints. The sequential composition of s_7 (10 and STN 9) with s_4 results in the constraints:

$$2.D_{mi} + d_{mi} + d_{2mi} \leq x_{i3} - x_1^7 \leq 2.D_{Mi} + d_{Mi} + d_{2Mi} \quad (19)$$

3.9 Proving the property

If all the scenarios are considered, the time lapse between the last execution of E (maintained) and the final state (three gears extended) may vary from the minimal value δ_m to the maximal one δ_M

$$\delta_m = \max_{i=1,3}(D_{mi} + d_{mi} + d_{2mi}) \quad (20)$$

$$\delta_M = \max_{i=1,3}(2.D_{Mi} + d_{Mi} + d_{2Mi}) \quad (21)$$

As the upper bound for d_{2Mi} is l_{0i}/a_i, if the maximal acceptable duration is Δ, then the property is verified for all values of the parameters D_{Mi}, d_{Mi}, l_{0i} and a_i such that:

$$\max_{i=1,3}(2.D_{Mi} + d_{Mi} + l_{0i}/a_i) \leq \Delta \quad (22)$$

4. CONCLUDING REMARKS

The important point to underline is that when classical approaches based on *model checking* allows to prove that some property is verified for a given set of values of the parameters, this approach, based on *theorem proving*, allows to derive the set of constraints which have to be verified by the parameters in order to turn true the property. The cost is that some symbolic calculus has to be done whereas model checking is based on automated tools. However, some parts of the method can be automated as p-invariant analysis. A tool for exhaustively deriving all the scenarios leading to a given partial marking is currently under development (Medjoudj *et al.*, 2004).

The presented approach is hybrid, even if it is based on a temporal abstraction of continuous dynamics. The reason is that the temporal abstraction is *symbolic* and not simply *numerical*. At the end of the calculus, the variables depicting the temporal abstractions are either replaced by numerical values or by symbolic expressions of some parameters, *taking into account any knowledge about the state of the system*. For instance, when the two variables d_{2mi} (lower bound) and d_{2Mi} (upper bound) are replaced, any knowledge about the current value of continuous variable l_i, when place p_{i2} receives a token, has to be used. When the concatenation of scenarios s_7 and s_4 is concerned, it is known that $l_i = l_{0i}$ and $d_{2mi} = d_{2Mi} = 0$. When scenario s_3 is concerned, it is known that $l_i = 0$ and $d_{2mi} = l_{0i}/b_i$ and $d_{2Mi} = l_{0i}/a_i$. Although the symbolic temporal expressions are the same (equations 19 and 12), after replacement of the symbolic variables the hybrid constraints are:

$$2.D_{mi} + d_{mi} \leq x_{i3} - x_1^7 \leq 2.D_{Mi} + d_{Mi} \quad (23)$$

and

$$2.D_{mi} + d_{mi} + l_{0i}/b_i \leq x_{i3} - x_1^3 \leq 2.D_{Mi} + d_{Mi} + l_{0i}/a_i \quad (24)$$

They are clearly different. *The difference between temporal and hybrid analysis is that in the second case the temporal constraints attached to the Petri nets depend on continuous variables and are dynamically computed.*

Finally, it is important to underline the fact that the approach favors *modularity*. Scenarios can be composed in sequential and parallel manners (see (Rivière, 2003)) without loss of partial order *i.e.* without adding spurious precedence constraints.

Acknowledgement: This work has been partially supported by the Network of Excellence HYCON.

REFERENCES

Boniol, F. and F. Carcenac (2002). Une étude de cas pour la vérification formelle de propriétés temporelles.
In: *Journées FAC 26-26 March 2002*. Toulouse, France, http://www.laas.fr/FERIA/SVF/.

Champagnat, R., R. Valette, J.C. Hochon and H. Pingaud (2001). Modeling, simulation and analysis of batch production systems. *Discrete Event Dynamic Systems: Theory and Application, Kluwer Academic Publishers* **11, n.1/2a, January/April**, 119–136.

Dechter, R., I. Meiri and J. Pearl (1991). Temporal constraint networks. *Artificial Intelligence* **49**, 61–95.

Girard, J-Y. (1987). Linear logic. *Theoretical Computer Science* **50**, 1–102.

Mancel, C., P. Lopez, N. Rivière and R. Valette (2002). Relationships between Petri nets and constraint graphs: application to manufacturing. In: *15th IFAC world congress*. Barcelona, Spain, 21-26 July. p. 634.

Medjoudj, M., S. Khalfaoui, H. Demmou and R. Valette (2004). A method for deriving feared scenarios in hybrid systems. In: *Probabilistic Safety Assessment and Management (PSAM 7 - ESREL 04)*. Berlin, Germany, 14-18 June.

Rivière, N. (2003). Modélisation et analyse temporelle par réseaux de Petri et logique linéaire. Phd thesis. Institut National des Sciences Appliquées, Toulouse, France.

STRQDS (2002). *Présentations d'études de cas*. October 4 2002. http://www.laas.fr/strqds.

Villani, E. (2004). Modelagem e análise de sistemas supervisórios híbridos. Phd thesis. Escola Politécnica da Universidade de São Paulo, Brazil.

Villani, E., J.C. Pascal, P.E. Miyagi and R. Valette (2003). Differential predicate transition Petri nets and objects, an aid for proving properties in hybrid systems. In: *ADHS 03, IFAC Conference on Analysis and Design of Hybrid Systems*. Saint-Malo, France, June 16-18. pp. 117–122.