

# A method for deriving feared scenarios in hybrid systems

Malika. Medjoudj<sup>1</sup>, Sarhane. Khalfaoui<sup>1,2</sup>, Hamid. Demmou<sup>1</sup>, Robert. Valette<sup>1</sup>

<sup>1</sup>LAAS-CNRS, 7 avenue du Colonel Roche,  
31077 Toulouse, FRANCE

<sup>2</sup>PSA Peugeot Citroën, 18 rue des fauvelles,  
92256 La Garenne Colombes, FRANCE

## Abstract

The long-term objective is to evaluate the dynamic reliability of mechatronic systems. We propose in this paper a new version of the algorithm that allows deriving critical scenarios from a Petri net model. It is more accurate because it takes into account some continuous aspects of the system. These scenarios characterise how the system leaves the normal operation to go to the feared state by determining the sequences of actions and state changes leading to a dangerous situations.

## 1 Introduction

Nowadays cars include more and more electronic and computing systems that enhance the engine performance, active security and diminish petrol consumption and air pollution. Nevertheless, this makes more complex safety analysis of such embedded systems composed of mechanic, hydraulic, electronic and computing parts, and called mechatronic systems. When studying safety of such systems, it is necessary to take into account in a realistic way the interactions existing between their physical parameters (for example: temperature, pressure, speed ...) and both functional and dysfunctional behaviour of its components. Classical methods of safety, as fault trees [1], are not sufficient to deal with this kind of complex and hybrid systems because they are inherently dynamic. Safety analysis of such systems must include timing considerations and the order of the events [2].

This paper presents an approach for a qualitative analysis of mechatronic systems safety from the dynamic reliability point of view [3]. It aims to characterise the feared scenarios at the early design stage of the system. The fact that feared scenarios are rare makes the simulation-based methods ineffective [4]. The hybrid aspect of mechatronic systems (both continuous and discrete features) leads us to choose a model that associates Petri nets and differential equations [5]. The Petri net model describes the operation modes, the failures and the reconfiguration

mechanisms. The differential equations represent the evolution of continuous variables of the energetic part of the system.

One way to avoid the combinatorial state space explosion is to directly use the Petri net model to extract the feared scenarios without generating the reachability graph. We use linear logic [8] to get a new representation (based on causality point of view) of the Petri net model, and then extract the scenarios from this new representation. The advantage is that with linear logic we can derive a partial order of transition firings and focus the search on the parts of the model that are interesting for safety analysis [9]. This approach is based on the equivalence of reachability in the Petri net and provability of a sequent in linear logic [10].

Our modelling approach has the advantage of clearly separating the continuous aspects from the discrete ones. This allows a logical analysis (using linear logic) of the causalities resulting from the state changes, based on the discrete aspect. Thanks to this analysis, and starting from a feared state, it is possible to go back through the chain of causality and to point out all the possible scenarios leading to a feared situation. Each scenario is given by a partial order between the events necessary to the occurrence of the feared event, unlike the fault trees that give a set of static combinations of partial states necessary to obtain the feared situation [6]. We have developed an algorithm that formalises a systematic approach for automatically deriving critical scenarios from the system model [7]. As this algorithm only operates on the discrete aspect of the model, scenarios which are inconsistent with the continuous dynamics have to be eliminated in a second step. In this paper, we propose a new version of the algorithm that takes into account the continuous aspect and specially thresholds attached to some transitions in the Petri net model. This permit to more precisely determine the exact conditions of the feared events occurrence: what make the system leave the normal behaviour and go to the feared one, and to characterize its side effects.

The method and the algorithm are illustrated on a case study: a two-tank regulation system. It is shown how a large number of inconsistent scenarios are no longer generated.

## 2 Method for deriving critical scenarios

We call scenario a set of events (here transition firings) leading from one partial state (here partial marking) to another one and verifying a partial order. As we have stated in the introduction, we assume that the system is made up of a set of components. A partial state is the conjunction of the states of a subset of these components.

**Definition:** *A partial order is defined by a directed graph  $(E, A)$  where the nodes  $E$  are a set of transition firings and the arcs  $A$  are pairs  $(t_i, t_j)$  such that  $t_i$  precedes  $t_j$  ( $t_i$  and  $t_j$  are transition firings).*

Starting from a partial knowledge of the scenario that leads to the feared partial state, we progressively enrich this knowledge by analysing the scenario and either introducing components states necessary to its occurrence or considering other

component states, which forbid it. In the first case, this is formalised by adding new tokens, which are necessary to fire some transitions within the scenario. In the second case, we add new tokens in order to fire transitions, which are in conflict with some transitions in the scenario.

This method is made up of two steps: a backward and a forward reasoning. The backward reasoning starts from the partial feared state in order to derive the events which are necessary to reach it and gives the last nominal states preceding the abnormal behaviour. The forward reasoning starts from these nominal states, enriches the scenario and points out the bifurcations between it and the normal operation. For both backward and forward reasoning, the starting point is a partial knowledge of the initial and final markings and the list of transition firings is unknown.

Before presenting the scenario derivation algorithm, we will first introduce the case study on which we will illustrate later the different steps of the algorithm.

### 3 Case study

#### 3.1 Presentation

The case study concerns a volume regulation system of two tanks (figure 1). It is made up of a computer, two pumps, three electrovalves, two volume sensors, the two regulated tanks (Tank1 and Tank2) and a third tank for draining. The demand is specified by a function of time (outgoing flowrates  $ds1(t)$  and  $ds2(t)$ ).

The volume of each tank ( $i$ ) must be kept within a given interval  $[V_{imin}, V_{imax}]$ . The volume is controlled by the computer, which decides, according to the values given by the volume sensors, to fill (or not) the concerned tank by opening (or not) the concerned electrovalve.

The control law of the computer is such that the electrovalve is closed when the volume of the controlled tank oversteps the upper limit  $V_{imax}$ . In the other hand, the computer commands the opening of the electrovalve each time the value of the volume in the controlled tank is lower than the limit  $V_{imin}$ . We distinguish two normal phases of the system, corresponding to the state of the electrovalve:

- A conjunction phase when the electrovalve is open. The volume in the tank is going up, no matter what is the value of the outgoing flowrate (the pump flowrate is much higher than the outgoing flowrate).
- A disjunction phase when the electrovalve is closed. The volume in the tank is decreasing.

This system must avoid the overflow of the tanks. A backup electrovalve is added to the system in order to drain the tanks in case of overflow. This third electrovalve is viewed as a shared resource between the two tanks, and it can be used to drain a unique tank at a time. When the volume of one tank oversteps the security limit ( $V_{il}$ ), the computer commands the opening of the backup electrovalve until the volume becomes lower than  $V_{imin}$ . As we focus our study on critical scenarios, and in

order to simplify the problem we consider that only the electrovalves can have failures. A typical failure of the electrovalves 1 and 2 corresponds to a blocked open state in which the electrovalve does not react to a closure command of the computer. These two electrovalves can be repaired after a failure occurrence. When the electrovalve 3 has a failure it is considered to be definitively out of service.

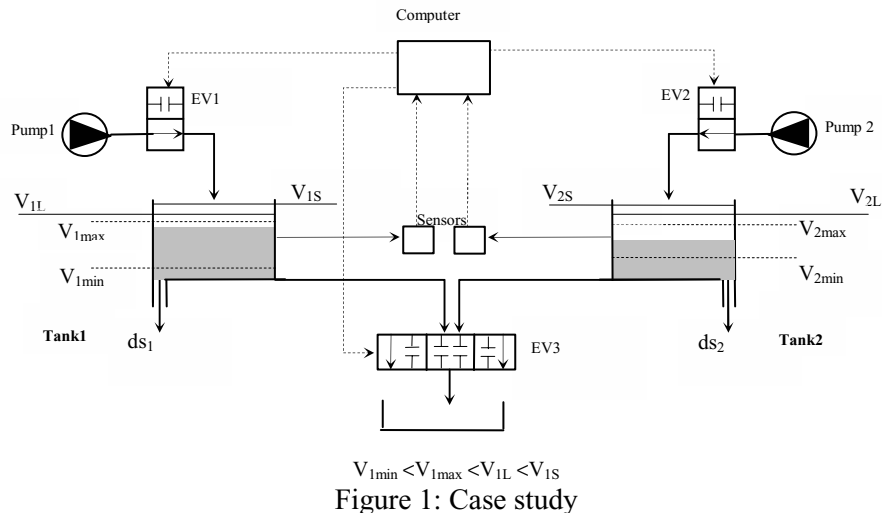


Figure 1: Case study

### 3.2 Petri Net Model

Place  $V1\_dec$  of the net in figure 2 represents the disjunction phase (the volume is decreasing); place  $V1\_cr$  represents the conjunction phase in which the volume is increasing. Place  $EV1\_OK$  corresponds to a state where the electrovalve 1 works. Transition  $t11$  represents the closing command of the electrovalve 1 when the volume oversteps  $V1max$ . Transition  $t12$  represents the opening command of the same electrovalve when the volume becomes lower than  $V1min$ . Transitions  $def1$  and  $rep1$  represent the fact that the electrovalve can stay blocked in an open state ( $def1$ ), and can be repaired ( $rep1$ ). Tank 2 is modelled in the same way. When the volume in the tank 1 oversteps the high security limit ( $V_{1L}$ ), and the backup electrovalve is available (place  $EV3\_OK$  is marked) then  $t14$  becomes fireable and the draining process of tank 1 can start via the backup electrovalve by marking place  $EV3\_oc1$ . The backup electrovalve is no longer available for use to drain tank 2, this corresponds to the place  $EV3\_OK$  empty. This phase lasts the time that it takes for the volume to reach the low threshold  $V1min$ . Then, the electrovalve 3 is released (place  $EV3\_OK$  is newly marked), and a conjunction phase is started again (place  $V1\_cr$  is marked) by firing transition  $t15$ . The electrovalve 3 can have a failure (modelled by transition  $def3$ ). In that case, place  $EV3\_HS$  is marked and the electrovalve is set out of order.

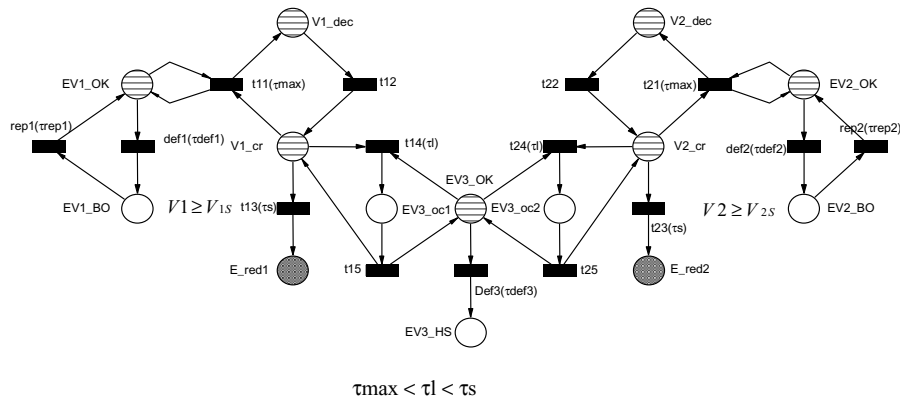


Figure 2. Petri net model of the regulation system

## 4 The Algorithm

As described in [7] the algorithm is based on different data structures and procedures implementing the backward and forward reasoning. In particular, these data structures handle sets of transitions, which can be in conflict in order to generate all the possible alternatives for the system behaviour.

The new version of the algorithm introduces conditions for transition firings. They are thresholds involving continuous variables. They are approximated by durations corresponding to the time for the system to reach the thresholds. It also introduces a new data structure (Lnft) that is a list of enabled or potentially enabled transitions that cannot be fired because they are in conflict with a transition that has to be fired before it (according to these new timing conditions). As a consequence, the scenarios that would have been generated are no longer constructed by this new version of the algorithm.

## 5 Illustration of the algorithm on the case study

Applied to the example, the old version of the algorithm generates 23 partial orders from which four corresponds to the feared scenarios for Tank 1, nine to its normal behaviour and ten are inconsistent with respect to the continuous behaviour. Using the new algorithm eliminates the ten of the inconsistent scenarios. As a matter of fact the consequence of the thresholds associated with transitions t11, t14 and t13, is that the transition t13 will be fired only if t11 and t14 are not enabled. This means that the feared scenarios are composed by fragments containing transitions in conflict with t11 and t14, and by the firing of t13.

For example the following scenario  $\{t13, def1, t23, def2\}$  given by the old version is not produced in the new version because transition t14 that is in conflict with t13 has an inferior threshold so it is fired before and forbids the firing of t13.

## 6 Conclusion

We have presented and illustrated an algorithm for deriving scenarios from a Petri net model. This new version of the algorithm by taking into account the continuous aspect avoid generating many inconsistent scenarios. It is a clear improvement but the work has to be continued because we still obtain too many scenarios. There are two main problems to be addressed: some derived scenarios are not minimal and some scenarios are redundant. For example in the previous system we have found four feared scenarios covering two minimal feared behaviours.

We are currently working on the notion of minimality for our scenarios, which in our context is expressed under the form of a linear logic sequent. This notion will be integrated in the algorithm in order to make more accurate the termination criteria.

### References

1. Lee, W. S.; Grosh, D. L.; Tillman, F. A., Lie, C. H., «Fault tree analysis, methods, and applications – A review », IEEE Transactions on Reliability, August 1, 1985; ISSN 0018-9529; r-34, page 194-203.
2. Chris J. GARRET, Sergio B. Guarro, George E. APOSTOLAKIS, « The Dynamic Flowgraph Methodology for Assessing the Dependability of Embedded Software Systems », IEEE Transactions On Systems, Man, and Cybernetics, Vol. 25, No. 5, May 1995.
3. F. Dufour, Y. Dutuit, “Dynamic Reliability: A new model”,  $\lambda\mu$ 13-ESREL2002 European Conférence, Lyon - France - 18 au 21 Mars 2002.
4. P.E. Labeau: “A Survey on Monte Carlo Estimation of Small Failure Risks in Dynamic Reliability”. In International Journal of Electronics and Communications, Vol. 52, pp. 205-211, 1998.
5. R. Champagnat, P. Esteban, H. Pingaud, R. Valette, “Modelling and simulation of a hybrid system through Pr/Tr PN DAE model”, ADPM'98 3rd International Conference on Automation of Mixed Processes, 19-20 March 1998, Reims, France p. 131-137.
6. H. Demmou, S. Khalfaoui, N. Rivière, E. Guilhem, “Extracting critical scenarios from a Petri net model using linear logic” Journal
7. S. Khalfaoui, H. Demmou, E. Guilhem, R. Valette “An algorithm for deriving critical scenarios in mechatronic systems” IEEE SMC (System Man and Cybernetic)2002, 6-9 october, Hammamet Tunisie
8. J.Y Girard, “Linear Logic”, Theoretical Computer Science, 50, 1987, p.1-102.
9. B. Pradin-Chézalviel, R. Valette, L.A. Künzle: Scenario duration characterization of t-timed Petri nets using linear logic, IEEE PNPM'99, 8th International Workshop on Petri Nets and Performance Models, Zaragoza, Spain, September 6-10, 1999, p.208-217.
10. S. Khalfaoui, E.Guilhem, H. Demmou, R. Valette, “Une méthode pour obtenir des scénarios critiques dans les systèmes mécatroniques”,  $\lambda\mu$ 13-ESREL2002 European Conférence, Lyon - France - 18 au 21 mars 2002.