Monitoring of Dynamic Process by Hybrid Automata

Fathi Karou Karoui, Hassane Alla, Abderazak Chatti

MONITORING OF DYNAMIC PROCESS BY HYBRID AUTOMATA

Mohamed Fathi Karoui[1,2], Hassane Alla[1] and Abderrazak Chatti[2])

[1]Gipsa-Lab, Département d'Automatique, 961, Rue de la houille blanche – BP- 46, 38402, Saint Martin d'hères.
[2]Institut National des Sciences Appliquées et de Technologie, Centre Urbain Nord BP 676 - 1080 Tunis Cedex.

ABSTRACT

In this paper we are going to present a monitoring approach of dynamic systems by using the hybrid automaton. At first we shall use the linear hybrid automatons, in continuation we are going to enrich our monitoring method by using the rectangular hybrid automatons. This work is based on the observation of the dynamic evolution of these systems, and activates an alarm if there is any infringement of the constraints which theirs are applied. The monitoring system, we propose, makes it possible to detect this infringement as soon as possible thanks to the reachability analysis. Our monitoring method will be applied to real physical systems.

Index Terms **-** Monitoring, linear and Rectangular hybrid automats, dynamic processes, Reachability analysis.

1. INTRODUCTION

Monitoring is an essential tool which accompanies henceforth every industrial system. Indeed the temporal constraints and the quality standards made the follow-up of the process evolution in real-time a necessity. It is very important to know if the manufacturing product will be finalized for the deadlines or not. Generally the used mechanism for monitoring the system constraints are based on watchdog which detect a fault if the dysfunction is produced early or late with respect to the time interval [14][15]. These Traditional techniques of monitoring wait for the end of the process execution to establish if yes or not there was dysfunction. The constraints applied in systems define what we call acceptable behaviour. It is necessary to establish a mechanism of monitoring which announces as soon as possible to the operator that the system deviates from its acceptable behaviour [1] [6] [17].
It is often the case between the theory and the practice there is frequently a divergence, these systems under real operation do not work as expected. Indeed a dysfunction can appear, the causes of this faulty operation are varied; occurrence of a failure, a bad decision of the control system or an erroneous indication of the sensors [10] [3].
The purpose of a monitoring system used on an industrial process, is to emit an alarm by analyzing the information sent out by the captors or signals coming from the command process (Figure 1). The issue of the monitoring of industrial systems has been dealt with in several works; both on the continuous systems [6] and on systems with discreet event and hybrid systems [13]. Very few works, however, are to be found on dynamic hybrid systems.

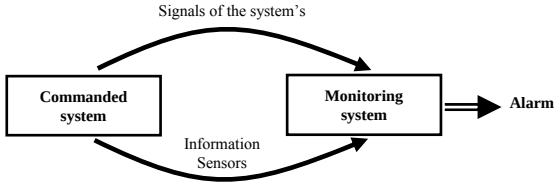


Figure 1. Commanded system of monitoring model

The major issue we are concerned with here consists in studying a system in real time that is able to develop in several different functioning processes. Each of these processes having a distinct evolution dynamics. A system's dynamics can be represented in various ways. More the dynamic is complex; the formal analysis possibilities are more weak.
In this paper we will present two approaches; in the first we consider that the system can evolve in different ways of functioning: The first one is the initial mode where the parameters are initialized to zero; the second is the normal mode where we have an evolution with the pre-established nominal parameters. From the normal state, several modes of dysfunctions are possible; each one has its own dynamics. This approach is based on linear hybrid automaton as modelling tool.

The second approach is based on the rectangular hybrid automaton (*RHA*), she allows us to model systems more complex than those who are modelled by the first approach.

2. MONITORING BY USING LINEAR HYBRID AUTOMATON

2.1. System behavior

We consider a monitoring system including several functioning ways; initial mode, normal mode and the mode *i* of dysfunction. Each mode is defined by a distinct dynamics. In figure 2, the mode *i* is represented in a generic way. We define variables *x* and *y* which are observable parameters, they are in the monitoring system and describe the state of the system. The variable *x* represents the duration of total execution of the process. The variable *y* reflects the state of the process's progress. The execution of the process is correctly made if; 1) the variable *x* must be included in the interval [α, β] (acceptable duration) and 2) the task was carried out $y = \delta$ (end value δ).

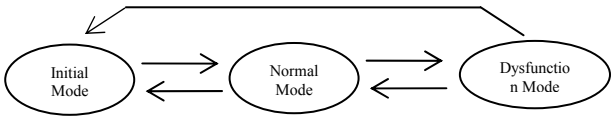


Figure 2. System in three states

During its evolution the process can commutate of the normal mode towards the dysfunction mode *i* according to the occurrence of the events (c_i, r_i). These events are supposed to be observable. This passage is also governed by constraints which evolve in permanence. These constraints are dependant of the various parameters. A return to the initial state is possible from the mode *i* if the process ended its execution while respecting the constraints on variables *x* and *y*. Figure 3 illustrate the behaviour of the process by a chronogram.

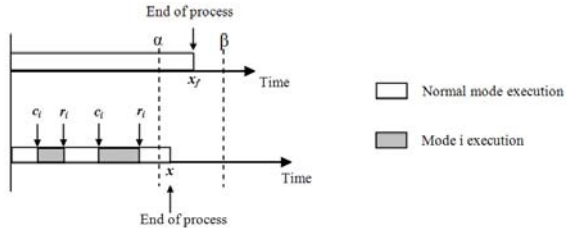


Figure 3. Process behavior

1.2. LHA, Model and analysis

There are several tools for the modelling of the dynamic discrete events systems. And some of them are automatons and Petri nets [7]. We chose the linear hybrid automaton because of its capacity of formal analysis [5] [9].
A hybrid automaton is a formal model which operates by alternation of continuous steps and discrete steps. In the progression of the continuous part, the state variables and the time evolve in a linear way. And in discrete part several discrete and instantaneous transitions can be crossed. Thus, the discrete changes are described by an automaton of finished state and the continuous dynamics by a set of variables and continuous equations.

A linear hybrid automaton is a 7-tuple H = (L, X, A, Σ, *dif*, *Inv*, l_0) [2] where:
- L: is a finite set of location,
- X: is a finite set of positive real-valued
- A: is a finite set of arcs. a = (l, δ, σ, R, l') $\in A$ is the arc between the locations l and l_0, with the guard δ, the label name σ and the set of stopwatches to reset R.
- Σ : is a finite set of labels,
- *dif*: function associating with each location $l \in L$ a set of continuous behaviours, *Dif*(l) : $\left.\frac{dx_i(u)}{du}\right|_t = x_i(t) = cste_l$
- *Inv*: maps an invariant to each location,
- l_0: is the initial location.

We present after some techniques which we shall use to calculate the spaces of the reachable states [16].
The state of an automaton is defined by the pair (L, E), L is a location of the automaton and E is its time state space, When the system reaches the location L_n the active counter have several values, all these values defines the time of the location L_n : E_n^a .
An automaton has two possibilities of evolution from the location L_n :
- Stay in the same location whereas the time passes by, the space of reachable state of this evolution is called continuous successor Suc_t.

$$E_n = Suc_t(E_n^a)$$

- Firing a transition a = (L_n , $g_{n,\, n+1}$, σ, R, L_{n+1}) $\in A$, All the reachable states since any state E_n is called discreet successor of the region E_n : Suc_d.

$$Suc_d(E_n) = E_n \wedge g_{n+1} \wedge R \qquad (1)$$

This analysis method is called forward analysis.

We also define the backward analysis method which allows calculating the predecessors of every region. The concept of continuous predecessor is dual with that of continuous successor. By letting the time progress any state E from which we can reach a given state Q is considered as a continuous predecessor of this state if we stay in the same location.

We note: $E = Pre_t(Q)$

Also the concept of discrete predecessor is dual with that of discrete successor.

1.3. Process modeling

To model the behaviour of the process of figure 2, we use a model of automaton who is composed of three states in

addition to the alarm mode [12]. To simplify the presentation, only one mode of dysfunction is considered. The generalization of the approach is one of our prospects for research.

The dynamics of the failing mode is λ_i belong to the interval $[\lambda_{min}, \lambda_{max}]$, with $\lambda_{max} \geq \lambda_{min} \geq 0$. The assumption that the dynamic ones are positive is a choice resulting from our experiment, the physical processes can be carried out more quickly or more slowly but seldom by reversing their direction of execution. If we consider the example of a valve through which a liquid flow, according to the opening of the valve the flow speed can increase or decrease. Nevertheless there is no theoretical constraint which prevents to have negative dynamics.

The event *b* represents the beginning of the process execution, and the event *d* represents his ending. The failing mode is presented by the c_i and r_i events, indeed event c_i brings the process towards the failing mode and event r_i brings back it towards the normal mode.

Variable *x* indicates time since the beginning of execution and variable y reflects the state of the ordered system.

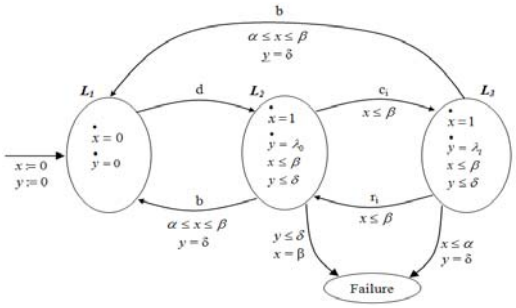


Figure 4. Automaton model of the process

1.4. Space of the reachable states

1.4.1. Forward analysis

The forward analysis allows calculating all the possible trajectories of the system, including those which lead to an alarm.

The clocks are initialized at the beginning, the space of time at the entrance of location L_2 is $E_2^a = \{x = y = 0\}$, the evolution of state (L_2, E_2^a) is given by using the forward analysis. E_2 is the discrete successor of E_2^a.

One notes $E_2 = Suc_d(E_2^a)$.This calculation is made by using the software PHAVER [8].

In our case one obtains:

$$0 \leq \frac{\lambda_i . x - y}{\lambda_i - \lambda_0} \leq \beta \ \wedge \ y \leq \delta \ \wedge \ 0 \leq x \leq \beta \qquad (2)$$

1.4.2. Backward analysis

To eliminate the trajectories which do not correspond to a correct execution, We carries out a back analysis starting from the area *D*, which represents the behaviour of the process executed correctly, in an other words who checks the final constraints described by the inequalities:

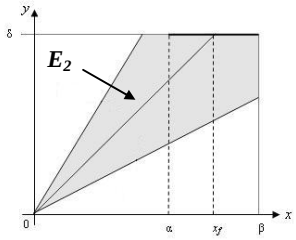
$$\alpha \leq x \leq \beta \ \text{ et } y = \delta. \qquad (3)$$

We calculate the space *A* which will enable us to reach the area *D* by the backward analysis method. It consists in reversing the automaton and making a forward analysis.

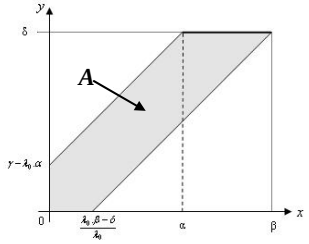
Space *A* (Fig. 4.b) is described by the following inequality:

$$\delta - \lambda_0 . \beta \leq y - \lambda_0 . x \leq \delta - \lambda_0 \alpha \qquad (4)$$

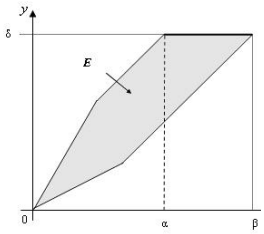
To obtain the space characterizing the correct evolution of the system, we calculate the area E which is the intersection of the areas E_2 and E'_2 (Fig. 4.c). $E = E_2 \cap A$



(a) Space state calculated by the forward analysis (E_2)



(b) Space state calculated by the backward analysis (*A*)



(c) Space states of the ordered process

Figure. 4: Space state

Any variation of the space E (Figure 4.c), will immediately involve an alarm activation.

3. MONITORING BY USING RECTANGULAR HYBRID AUTOMATON

3.1. RHA, Model and analysis

The continuous dynamics analysis of the dynamic hybrid systems is complex. It is why we concentrated on the study of a subclass of hybrid dynamic systems, modelled by rectangular hybrid automatons, checking numbers of conditions. This model allows approximating the dynamics of the continuous behaviour of the system, under the use of conditions dealing with rectangular flows of the shape $\dot{x} \in [\alpha, \beta]$.

Definition 1: A Rectangular hybrid automaton is a 7-tuple (*Q, X, Σ, E, inv, flux, init, M*) [10] where:

- $Q = \{q_1, ..., q_k\}$ is finite set of location,
- *X* is a finite set of real variables,

- Σ is a whole of events,
- $E \subseteq Q \times \Sigma \times Rect(X) \times Rect(X) \times 2^X \times Q$ is a finite set of transitions. A transition (q, σ, g, r, R, q') corresponds to a change of top of q to q', on the occurrence of the event σ and under condition $v \in [g]$,, where vector v corresponds to the current values of the variables of X,
- $inv : Q \rightarrow Rect(X)$ is a function which associates with every location $q \in Q$ a rectangular constraint for each variable $x_i \in X$,
- $flux : Q \rightarrow Rect(\dot{X})$ is the function which assigns to each location a representation for the continuous evolution,
- $init \subseteq Q \times Rect(X)$ indicate the initial condition of the automaton,
- $M \subseteq Q$ corresponds to the whole of the marked tops of the automaton.

As mentioned above in this paper, generally the study of a system modelled by a rectangular hybrid automaton is based on the reachability analysis of the automatons states. To know if a region R is reachable since a region R_0, two methods can be used; the forward analysis and the backward analysis [11].
A region (or a symbolic state) of an *RHA* is represented by a pair $\langle q, z \rangle$, where q corresponds to a location of automaton and z a region of the space of continuous state, represented by a polyhedron.
We shall demonstrate afterward that the intersection of trajectories obtained with the forward analysis method and those obtained with the backward analysis will give all the trajectories which characterize the normal evolution of the functioning of system.

3.2 Process modelling

The general structure of a monitoring model is a rectangular hybrid automaton given in Figure 5. In this figure the modes correspond to the normal functioning. There is a guard between every location of a normal functioning towards the failure location; this guard is given by the expression: $\forall i \in [1, n]\ Gd_i = I_i$, With I_i the invariant concerning every location i.

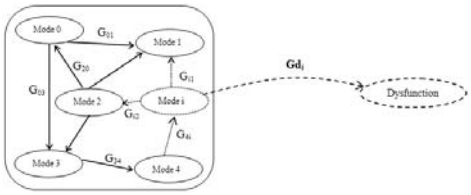


Figure 5. General structure of monitoring model

Definition 2: [4] the system is ''live'' if it can always execute an event. When there is a set of unmarked states that forms a strongly connected component (*i.e.* these states are reachable from one another), but with no transition going out of the set. If the system enters this set of states, then we get what is called a livelock.

Definition 3: A monitoring model is a monitoring system if it is in a livelock evolution.

Let be E_{Av} the space of the states calculated by the forward analysis method, this region is defined by the pair $\langle q_A, z_A \rangle$. And let E_{Nor} be the space of states characterizing the normal functioning, in this space the *RHA* model is in a livelock. As space E_{Av} characterizes all the possible trajectories starting from the initial state $\langle q_0, z_0 \rangle$, the following relation can be written: $E_{Nor} \subseteq E_{Av}$, which is illustrated by the Figure 6.

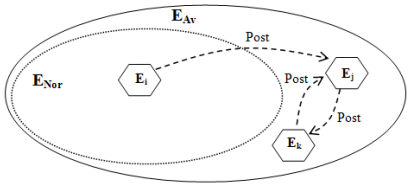


Figure. 6. Space state by the forward analysis and spaces state of normal functioning,

Hypothesis: There is a region $E_i \in \{E_{Nor}, E_{Av}\}$ such as post $(E_i) = E_j \notin E_{Nor}$, *i.e.* one of the E_i successors is not included in region E_{nor}. The successor of the region E_j will be the region E_k and the successor of this one will be E_j. It is a system blocked in several states only. According to this hypothesis there is no trajectory bringing the system of the state E_j to a state of the normal functioning E_{Nor}. For this fact the system is not considered as in a livelock.
As proved above, space of states calculated by the forward analysis is not sufficient to characterize the space of the states of the normal functioning. The monitoring model *RHA* is not in livelock evolution, sooner or later the system will enter a restricted sub-space where it cannot go out of it any more (Fig. 7).

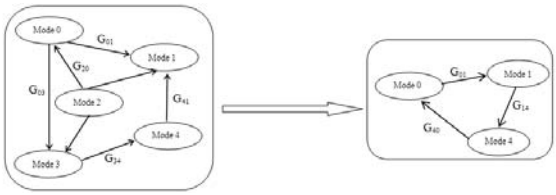


Figure 7. Blocking of the RHA in a subspace

In this sub-space, transitions cannot be crossed, and thus crates behaviour different from the normal functioning.
The space of states calculated by the backward analysis method is determined from a final state belonging to the normal functioning and by inverting the transitions arcs of the *RHA* model.

Let E_F the region defined by $\langle q_f, z_f \rangle$ (Fig.8); E_F is the final state of normal functioning. Predecessors of E_F belong to E_{AR}. E_{AR} is the space of states calculated by the backward analysis method. Hypothesis: one of the predecessors of E_f is the region E_i and E_0 the initialized start space $\langle q_0, z_0 \rangle$ of the normal functioning is not a predecessor of E_i. This hypothesis allows us to conclude that the space obtained

by the backward analysis method is not enough to define the space of the normal functioning.
The space of the normal functioning state is defined such as every region in this space it is at the same time a successor and a predecessor of another region belonging to this space. The E_{Nor} space calculated by the relation $E_{Nor} = E_{AV} \cap E_{AR}$ satisfies this condition.

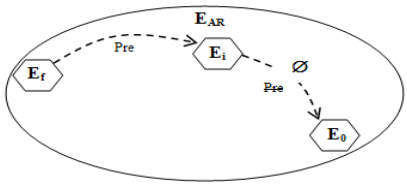


Figure 8. Spaces states defined by the backward analysis.

Property 1*: *RHA is in a livelock evolution if the space of the forward analysis and the backward analysis are equal.

***Property 2*:** A monitoring system is obtained from the monitoring model by replacing all the invariants departure with the calculated spaces. These spaces are determined by making the space's intersection of the forward analysis method and the backward analysis method.

Every region $\langle q_i, z_i \rangle \in E_{Nor}$ is at the same time a successor of $\langle q_0, z_0 \rangle$ and a predecessor of $\langle q_f, z_f \rangle$. There is always a trajectory L which can bring the system in any state $\langle q_i, z_i \rangle$ starting from the initial state $\langle q_0, z_0 \rangle$.
The whole of transitions S and the set L of a finite trajectories which brings from an initial state $\langle q_0, z_0 \rangle$ to any state $\langle q_i, z_i \rangle$ belonging to E_{Nor} space is in a livelock functioning.

4. ILLUSTRATIVE EXAMPLE

In this section, we'll show the relevance of our modelling and monitoring approach by an isolated transmission system. The function of this system is to collect data and then to send them. The sending of data is dependant on the power consumption as well as the emission's rate. The transmission mechanism can adopt two states of functioning; the state "*On*" and the state "*Off*". During state "*Off*" the system puts itself in loading mode and during the sending phase it is in energy consumption mode. The model *RHA* of the system is developed in Figure 8.
To characterize the functioning of the system, we use three variables h, x and y. h variable represents time of emission; indeed we consider that data emission is limited on time. Variable x represents the energy reserve contained in the system, to transmit suitably data; the system has to have an included energy stock between x_{min} and x_{max}. The values of this interval are deduced from the physical constraints of the system. x_{min} represents the minimal value of the energy below which the system would be unable to work. x_{max} indicates the maximum limit of energy which the system could store. Variable y expresses the emitted quantity of data.
The dynamic of the variables h, x and y are:
- $\dot{h}$: it represents the operating condition of the system "*On*" or "*Off*".
- $\dot{x}$: represents the rate of energy consumption; we adopt a positive dynamics during phase "*Off*" of the system and negative dynamics are used to reflect the decrease of energy quantity during the functioning.
- $\dot{y}$: represents the rate of data emission.

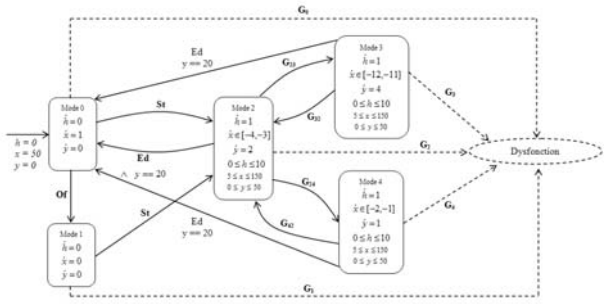
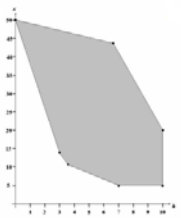


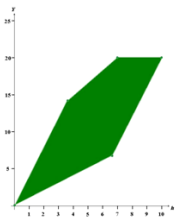
Figure 9. RHA model of the transmission system

The transmission system can evolve in 5 ways of functioning. Initially the system in mode 0 is loading, from there it can evolve in mode 1 when the quantity of stored energy reaches the authorized superior limit or towards the others modes (mode 2, mode 3 and mode 4). The function assigned to this system is to emit a series of data of size y=20 in a period which does not exceed h = 10 *t.u.* The monitoring system modelled by the Fig. 9 enables us to indicate as soon as possible if transmission is going to be made for the indicated deadlines.

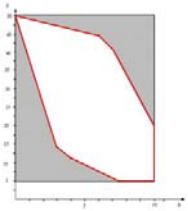
The reachable spaces of variables x and y are respectively represented in Figure10.a and in Figure 10.b. These spaces were calculated by the software PHAVER [8]. Any trajectory included in these spaces satisfies the constraints of data emission. And any trajectory not belonging to the reachable spaces of x and y is a trajectory which brings the system to a dysfunction.



(a) Parameter x



(b) Parameter y

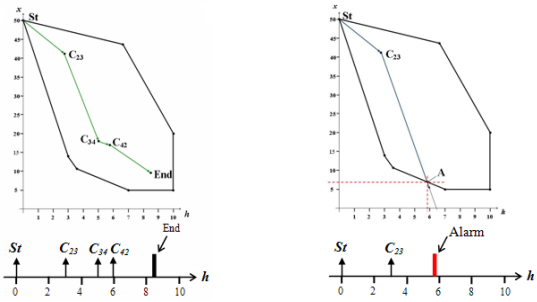


(c) superposition of both space of parameter x

Figure 10. Reachable space state

In the Figure 10.c, we stack two spaces of state; the first one, the reachable states of the variable *x* and the second (frame) representing the total space of evolution of the variable *x*. This figure illustrates well the relevance of our approach. Indeed the space which we calculate is much lower than that established by monitoring system already in place.

The example of the first trajectory defined by the following orders (*St, C_{23}, C_{34}, C_{42}*) is illustrated in Fig. 8.d. The broadcast of data is carried out without violating the invariants. On the other hand the trajectory 2 (Figure 11.a) defined by the sequence of order (*St, C_{23}*) brings to a release of alarm in the point A with coordinates (5, 8, 7).



(a) normal operation (b) dysfunction

Figure 10. Example of operation

By using the traditional techniques of monitoring, the alarm starts for $h = 6$ *t.u.* The method we propose has allowed in this case time-saver of 0,2 *t.u.*

5. CONCLUSION

In this paper we have proposed two monitoring approach of dynamic systems. The monitoring model of this system is based on hybrid automaton. It takes into account the dynamic changes which can show up during the execution of the process while keeping dominant events. The summits of the automaton represent the various dynamics the commanded system can have; the transition between modes of function is synchronized by events related to these various dynamic. The authorized behavior of the system is controlled by variables to which constraints are applied. These constraints expressed by inequalities define the acceptable space of the ordered system evolution.

6. REFERENCES

[1] Allahham A and alla, H. Monitoring of timed discrete events systems: Application to manufacturing systems. In The 32nd Annual conference of IEEE Industrial Electronics Society, Paris, 2006.

[2] Alur R, C. Courcoubetis, N. Halbwachs, T. A. Henzingerd, P. H. Hod, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 138(1), 1995

[3] Ayari I., Chatti A., Reactive Control Using Behavior Modelling of a Mobile Robot , International Journal of Computers Communications & Control, ISSN 1841-9836, 2(3):217-228, 2007.

[4] Cassandras C.G., S. Lafortune, Introduction to discrete event systems, Springer, 2008.

[5] Cassez, F. and Larsen, K. (2000). The impressive power of s topwatch. Number 1877, pages 38{152. Lecture Notes in Computer Science, Springer-Verlag

[6] Cocquempot, V., Mezyani, T. E., and Staroswieckiy, M. (2004). Fault detection and isolation for hybrid systems using structured parity residuals. In Proceeding of Asian Control Conference, ASCC'04.

[7] David R. and Alla H., Du grafcet au reseaux de Petri. Hermes, Paris 1995.

[8] Frehse, G. (2005). Phaver: Algorithmic veri_cation of hybrid systems past hytech. In Proceedings of the Fifth International Workshop on Hybrid Systems : Computation and Control, pages 258-273

[9] Henzinger, T. (1996). The theory of hybrid automata. In proceeding of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS'96, pages 278-292.

[10] Henzinger T.A. P.W. Kopke, A.Puri, V. Varaiya, What's decidable about hybrid automata? Journal of Computer Sciences, (57), 1998

[11] Karoui, M. F., Alla H. and other. (2010a) Monitoring of dynamic processes by rectangular hybrid automata, *Nonlinear Analysis: Hybrid Systems*, doi:10.1016/j.nahs

[12] Karoui, M. F., Alla H. and other (2010b) Surveillance des processus dynamiques par automates hybrides linéaires, Conférence Internationale Francophone d'Automatique (CIFA), papier n°187.

[13] Lunze J, Diagnosis of quantised systems by means of timed discrete-event re- presentations. Lecture notes in computer science, Springer, Berlin, 2000

[14] Pandalai D. N. and L. E. Holloway. Template languages for fault monitoring of timed discrete event processes. IEEE Transactions On Automatic Control, 45(5), May 2000.

[15] Srinivasan V.S. and M.A. Jafari. Fault detection/monitoring using time petri nets. IEEE Transactions on Systems, Man and Cybernetics, 23:1155–1162, 1993.

[16] Sava, A. (2001). Sur la synthèse de la commande des systèmes à événements discrets temporises. PhD thesis, Institute National Polytechnique de Grenoble, INPG.

[17] Tripakis, S. (2002). Fault diagnosis for timed automata. Proceeding 7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT'02), 2791 of Lecture Notes in Computer Science: 205 - 224.