ISSN 2229-5518

A Monitoring System For a sub-Class Of Hybrid Systems

B. Tolbi and H.Tebbikh
LAIG, University of Guelma, BP401 Guelma, Algeria.

Abstract—*In this paper, an approach for synthesizing a monitoring system to insure the detection of interruption, permanent and/or intermittent faults in a sub-class of hybrid systems namely continuous flow systems, is presented, this approach is based on system modeling using hybrid automaton with stopwatch, the faults detection is guaranteed by timer violation. The timer calculates the elapsed time from the beginning of operating cycle of the system. The obtained results show that the monitoring system is able to detect rapidly the considered types of faults. A classical example is dedicated to illustrate our approach; the results obtained confirm the effectiveness of the proposed work.*

Index Terms— Continuous Flow Systems, Faults modeling, Monitoring system, Hybrid Automata.

1 INTRODUCTION

HIBRID Dynamic Systems (HDSs) are dynamic systems integrating explicitly and simultaneously continuous and discrete event systems (DESs), which require for their description the use of a continuous model, and a discrete event model [16]. In this paper, a particular class of HDSs, which are Continuous Flow Systems (CFS) 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.

The hybrid character of a system can owe either to the system itself or to a discrete controller applied to a continuous system. Several works have been devoted to the modeling of HDSs. The most known model of this category is hybrid automata (HA). This model presents a lot of advantages. The most important is that it combines, explicitly, the basic model of continuous systems, which are differential equations, with the basic model of discrete event systems. The model-based monitoring approaches are relied on the comparison of the expected behavior of the system (described by model) and the actual observations. So, a fault occurrence is declared when some discrepancies occur between the two mentioned outputs.

Monitoring methods for HDSs are classified in two categories; non-model-based methods and model-based methods. The non-model-based approaches are highly dependent on the heuristic rules and require the process history. In this article, we focus on the second category in which the normal behavior of the system is modeled. These monitoring methods are based on models of normal function and/or the malfunction of system which are found in the literatures [1]-[7].

The most of works are not based on time aspect of faults and operating system regardless of the normal functioning or faulty functioning. For this we aim to develop a new and more powerful detection model based on time aspect. This paper proposes a monitoring approach for permanent and/or intermittent faults in sub-class of HDSs, whose principle is based on system modeling using hybrid automaton with stopwatch this model is proposed with two aspects: data acquisition and fault detection. The data collected during the actual operation of the system (in presence of faults) and they will be compared with normal operation.

In permanent faults, the passage towards a state of incipient fault is due to the occurrence of a fault event. Then the system will move to other states of dysfunction. In case of intermittent faults, the system can return to a normal operating state after occurrence of an event back in normal operation. In this work, we focus on two types of faults which are defined as follows.

2 FAULTS IN HDSS

Fault modeling involves the acquisition for prior knowledge of faults to be detected. We note that a system fault is a faulty state, while a failure or a fault source can lead to a faulty state. In the context of HDSs, the occurrence of a fault is, also, the passage towards a faulty state. This passage can be modeled by a transition on a fault, if we consider models-based events [8], [9]. If we consider a state-based model [10], a partition of system states in nominal conditions and faulty states is first established. In this last case, a system is declared faulty if it reaches a malfunction state.

Permanent faults are defined as malfunctions of components that need to be replaced or repaired. Therefore, a fault is permanent if recovery occurs after the repair or replacement of the faulty component. For example: The final judgment of the pump in Figure 1.

An intermittent fault is usually the result of a partial and gradual degradation of a system component, can lead to permanent fault. In this case, the system can regain its nominal operation after occurrence of the event. Therefore, fault is intermittent if the recovery occurs automatically (without resorting to replace the components). For example, valve blocking in opening or closing mode (Figure 1).

The difference between the above fault types is that the intermittent fault leads the system to switch between the offending state and the normal state (no faults), however, the permanent faults are always associated with events of overlap and the system cannot automatically switch from fault state to the normal state.

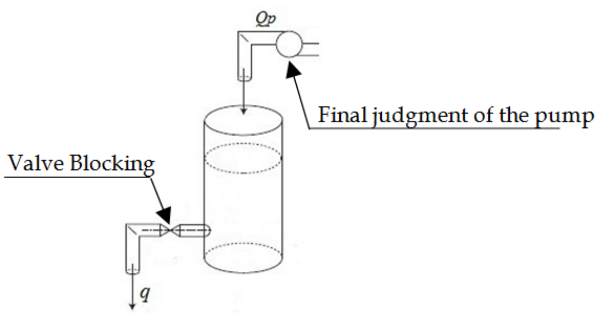

Fig. 1. Filling System.

3 Monitoring Approach

When The monitoring approaches in which the abstraction of continuous dynamics of HDS have been utilized (through the use of DES models) lead to a considerable loss of information and so are not reliable for the fault detection. In the other words, in some cases, failed behavior manifested by a trajectories deviation of continuous dynamics of system. Therefore, using the diagnostic approach based on a purely discrete abstraction of the system evolution is inadequate for the fault detection purposes.

To overcome this problem, a monitoring approach is proposed in this article based on the work presented by Karoui [11] and Derbel [12] illustrated in Figures 2, 3 and 4. Our goal is to develop a new approach for monitoring purposes utilizing the stopwatch hybrid automaton model (SHA). Figure 5 illustrates the overall scheme of our proposed monitoring approach. This approach takes into consideration the temporal time aspect in the evolution of HDS to be monitored. It relies on the use of a complete model of the system in the normal operation in the form of a hybrid automaton (HA). HA allow both modeling of discrete and continuous parameters that make up the HDSs.

The continuous component is described by a set of ordinary differential equations, and discrete component is described by a finite automaton. This model describes the time evolution of system events through the use of a set of timers. These fictitious variables are used to abstract the interactions between the continuous dynamics (variables) and discrete dynamics (events) of a system, using a set of temporal constraints (guards) conditioning the continuous evolutions and discrete evolutions.

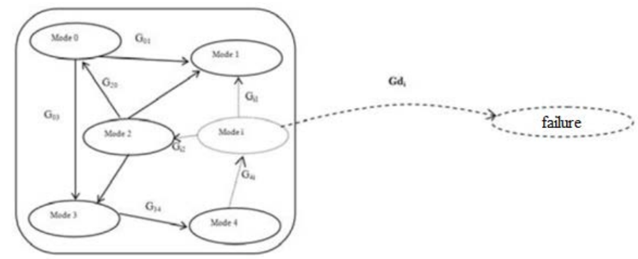

Fig. 2. General Structure Of A Monitoring Model.

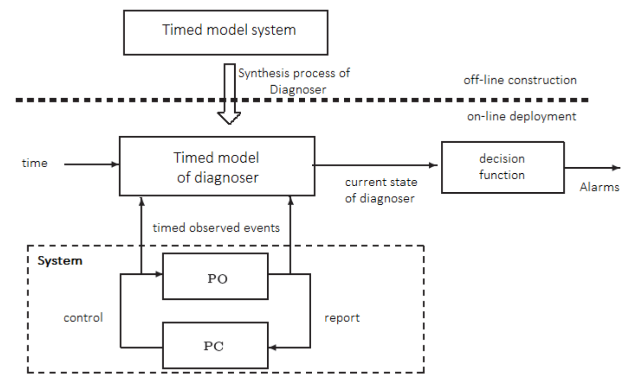

Fig. 3. Overall Approach Of Diagnostic Based On Timed Automata.

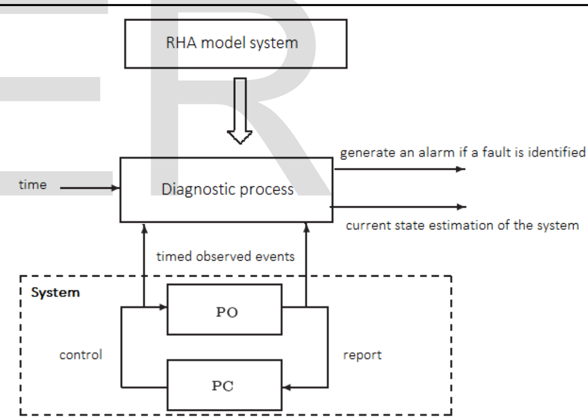

Fig. 4. Diagram Of On-Line Diagnostic Based On Rectangular Hybrid Automata.

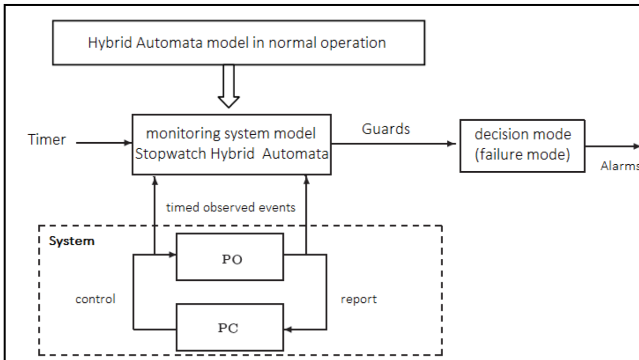

Fig. 5. Overall Scheme Of Our Proposed Monitoring Approach.

In the sequel, first, some required definitions are presented; then, the developed monitoring algorithm is detailed.

ISSN 2229-5518

Definition 1 (Hybrid automaton) A linear hybrid automaton [13] with stopwatch is 8-uplet H = (S, X, T, Σ, t, dif, Inv, S_0) in which the following variables are incorporated:

♦ *S:* the finite set of locations (also known as localities, situations);

♦ *X:* the finite set of real variables (continuous state vector components);

♦ *T:* the finite set of transitions. We note that a=(s,g,σ,R,s') $\in T$ where s is a location source, g is the guard, σ is the event associated, R is the assignment and s' is the destination location;

♦ *t:* the stopwatch timer which calculates the time at each operating cycle;

♦ *Σ:* finite set of labels (i.e. set of event-actions related to the transitions crossing);

♦ *Dif:* the function associating with every location s ∈ S the set of continuous behavior (also called as activities) $Dif(s)$ calculated as below:

$$\left.\frac{dx_i(u)}{du}\right|_t = x_i(t) = cste_l$$

♦ *Inv:* the function associating with every location $s \in S$ an invariant $inv(s)$;

♦ $S_0 \in S$: the initial location.

Definition 2 Assume that ev_j, $j \in \{1,...N\}$ is the set of all related events, where N represents the number of events in the system. We consider t as a timer that calculates the elapsed time from the beginning of the operating cycle of the system. This timer reset to zero after each cycle execution. Also assume that T_j, $j \in \{1,...N\}$ is a set of all occurrence times associated with each ev_j, According to the mentioned assumptions, we can derive the following condition:

If the event ev_j appeare when $t = T_j$, then the system is in normal operation,
Else, the timer indicates a violation of specific time associated with each ev_j.

Example 1 Let j=2, therefore they are: Two events, ev_1 and ev_2; Two moments of occurrence, T_1 and T_2.

If ev_1 appeared when $t = T_1$ and ev_2 appeared when $t = T_2$, the system works normally.
Else, the timer indicates a violation of the specific time associated with ev_1 and/or ev_2.

Definition 3 Assume that $X_i, i \in \{1,...n\}$ is the set of all state variables of continuous dynamics of HDS, where n is the variables number, $M_i, i \in \{1,...m\}$ is the finite set of modes (also called locations) and m is the number of modes in normal operation. Crossing conditions fixed in the HA are calculated using $x_i \in X_i$, which is an external variable of HA. We consider $T_s, s \in \{1,...S\}$ as the set of intervals in which S is the number of occurrence intervals of modes $M_i, i \in \{1,...m\}$ and assume that T_{d_i} is the occurrence interval of the fault d_i, According to the mentioned assumptions, we can derive the following condition:

If the condition C_i on x_i that keeps the input transition associated with each mode M_i is true ($t \in T_s$), then the system normally switches from the previous mode to the current mode,
Else, the timer indicates a violation of the specific time for each C_i that control the input transition of the current mode.

Example 2 Let m=2 and i=2, C ($x_1 \geq 0.5$) and the occurrence interval of the mode M_2 is $[50\ \ 60]$ therefore:

If $(x_1 \geq 0.5)$ and $t \in [50\ \ 60]$, the system switches from the mode M_1 to the current mode M_2, else, the timer indicates a violation of the specific time $[50\ \ 60]$ on the condition ($x_1 \geq 0.5$) which keeps the input of the current transition mode M_2.

From the last two definitions, we distinguished input transitions guards associated with each mode M_i. Accordingly, the monitoring algorithm can be developed as follows. The monitoring process is formally described in Algorithm 1. We begin by initializing the timer t then run the SHA. The variable t is used to measure the time elapsed between the occurrence time and disappearance time of the observable event. It can also measure the length of stay in each mode. This variable is reset after each execution cycle. This algorithm could be simplified as the architecture presented in Figure 7.

Algorithm 1 proposed monitoring Algorithm

Inputs: x_i
Outputs: M_i, t of timer

```
«INITIALIZATION», time initialization, t=0 ;
Loading data in SHA
If«VERIFICATION_CONDITIONS», the condition C_i do
  verification of timer t;
  If the timer t ∈ T_s do
    system normally switches from M_i to M_{i+1} ;
    Else if the guard g = −C_i is true when t ∈ T_s do
      the system switches to the malfunctioning mode M_{m+1} ;
        announce the occurrence of a fault d_i ;
        until the fault d_i disappeared do
              If t ∈ T_{d_i}
                 return «VERIFICATION_CONDITIONS»;
              else
                 End of the operating cycle;
              End
        until end
   else
   announce « normal », the system operates in the normal mode;
```

```
        end
    end
else
    the operating system staying in the normal mode M_i ;
End.
```

In the following, a hydraulic system has been used as s dedicated example to illustrate the process implementation.

4 Ellustrative Example

We consider the hydraulic system [14] depicted in Figure 6. It consists of [6]:

• Two tanks R_1 and R_2, with cross sections $S_1 = S_2 = S$;

• Four cylindrical pipes C_1, C_2, C_3 and C_4 have the same cross section A. the two tanks are connected by pipes C_2 and C_3 placed respectively at the levels $h_{12}= 0m$ and $h_{12} = 0.5m$. The pipes C_2 and C_3 are equipped with V_2 and V_3 valves.

• Two valves V_1 and V_4 allowing liquid evacuation for the use;

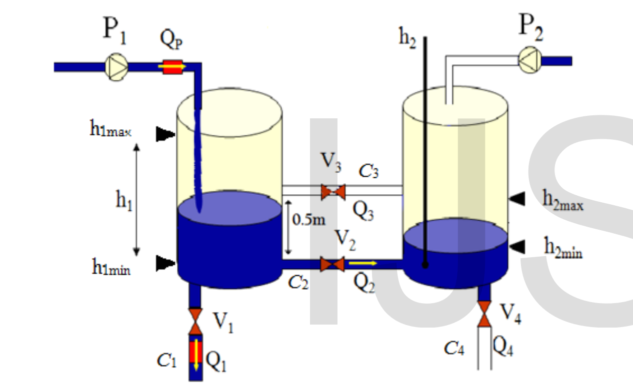

Fig. 6. Hydraulic System.

• Two pumps P_1 and P_2 have the same flow rate Q_p;

• Four sensors: two sensors measure the flow rates Q_p and QV_1 (flow rate through the valve V_1) and the other ones measure h_1 and h_2 that are the height of liquid in tanks R_1 and R_2 respectively.

Remark 1 To simplify the study, we consider that the valves V_1, V_2 and V_3 are kept constantly open.

4.1 System Modeling During Normal Operation

The modeling is an important step to understand the system behaviors, we use hybrid automaton which represents an extended version of finite state automaton associated with differential equations. Thus, the overall sate of a hybrid automaton, at a given time is defined by a pair (q, x) in which q represents the situation (discrete state) and x the state vector (in the sense of continuous). This global state changes for two reasons:

♦ The crossing of discrete transition is changed abruptly or directly by the evolution of continuous state. This crossing happens when an appropriate event occurs and/or if a condition becomes true;

♦ The temporal evolution that affects x following the differential equation associated to the current situation. This situation remains unchanged.

Flow rates expressions given in [6] and [14] by *Torricelli law* are:

$$\begin{cases} Q_1(t) = & S_1\sqrt{2g.h_1(t)} \quad ; \\ Q_2(t) = & A_2\ sign\left(h_1(t)-h_2(t)\right).\sqrt{2g.\left|h_1(t)-h_2(t)\right|} \quad ; \\ Q_4(t) = & S_4\sqrt{2g.h_2(t)} \quad . \end{cases} \tag{1}$$

Q_3 could be calculated using three expressions depending on liquid level in the tanks R_1 and R_2

$$Q_3 = \begin{cases} A_3\sqrt{2g.\left(h_1(t)-h_{12}(t)\right)} \quad , & if\ h_1 > h_{12}\ \ and\ \ h_2 < h_{12} \\ -A_3\sqrt{2g.\left(h_2(t)-h_{12}(t)\right)} \quad , & if\ h_1 < h_{12}\ \ and\ \ h_2 > h_{12} \\ A_3 sign\left(h_1(t)-h_2(t)\right)\sqrt{2g.\left|h_1(t)-h_2(t)\right|} \quad , & if\ h_1 > h_{12}\ \ and\ \ h_2 > h_{12} \end{cases}$$

To simplify writing, we rewrite Q_3 by the following expression:

$$Q_3 = A_3.sign\left(H_1(h_1)-H_2(h_2)\right)\sqrt{2g.\left|H_1(h_1)-H_2(h_2)\right|} \tag{2}$$

Where H_1 and H_2 are respectively the non-linear functions of h_1 and h_2

$$H_1\left(h_1\right) = \begin{cases} 0 & if\ h_1 < h_{12} \\ h_1 - h_{12} & if\ h_1 \ge h_{12} \end{cases} \tag{3}$$

$$H_2\left(h_2\right) = \begin{cases} 0 & if\ h_2 < h \\ h_2 - h_{12} & if\ h_2 \ge h \end{cases} \tag{4}$$

After developing and linearizing, the following equations can be derived:

$$\begin{cases} Q_1 = A.\sqrt{2g}.\sqrt{h_1} \\ Q_2 = A.\sqrt{2g}.\sqrt{\left|h_1 - h_2\right|} \\ Q_3 = A.\sqrt{2g}.\sqrt{\left|h_1 - h_{12}\right|} \\ Q_4 = A.\sqrt{2g}.\sqrt{h_2} \end{cases} \tag{5}$$

Table 1 shows the probable events (spontaneous or controlled) that may be happened [15].

We consider the following specifications:

♦ The valves V_1, V_2 and V_3 are always open,

♦ The pump is controlled to maintain h_2 in a fixed interval, the

TABLE 1
CONTROLS AND SPONTANEOUS EVENTS GENERATED BY THE SYSTEM

Notation	Description
ev_1	Event according to opening of valve V_4
ev_2	Event according to closing of valve V_4
$h_1 \geq h_{12}$	Level threshold h_{12} crossed by h_1 in ascending order
$h_1 \leq h_{12}$	Level threshold h_{12} crossed by h_1 in descending order

pump is switched on when $h_2 = h_{2max}$ (h_{2max} = 0.1m). The pump is stopped when $h_2 = h_{2min}$ (h_{2min} = 0.2m).

♦ The valve V_4 can be opened by the operator, but the action is only performed if the liquid level in the tank R_1 is greater than h_{12}. The valve V_4 is held closed when the liquid level in the tank R_1 is less than h_{12}.

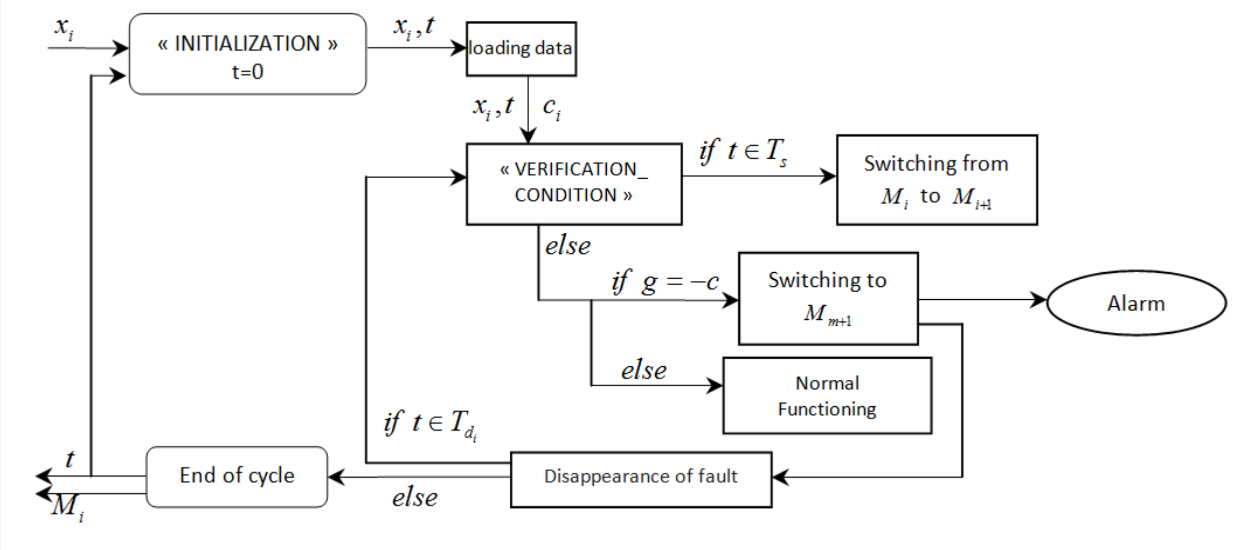

Fig. 7. Algorithm Architecture.

In normal operation, only two discrete states are considered: the state of the C3 pipe that can take the terms V (The two levels h_1 and h_2 are lower than h_{12}) or P (at least one level h_1 or h_2 is greater than h_{12}); and the state of the valve V_4 which can take the modality O (opened) or *F* (closed). The initial state corresponds to discrete state $q_0 = (V;F)$ and the continuous state $x_0=[0.4\ 0]^T$. Thus, the system initially starts with the levels h_1 and h_2 greater than h_{12}.

Remark 2 We consider a safety threshold: $h=h_{1min}$ (h_{1min} =0.001m) at which the pump should be started and superior threshold $h_1=h_{1max}$ (h_{1max} = 2cm) above, the pump should be stopped.

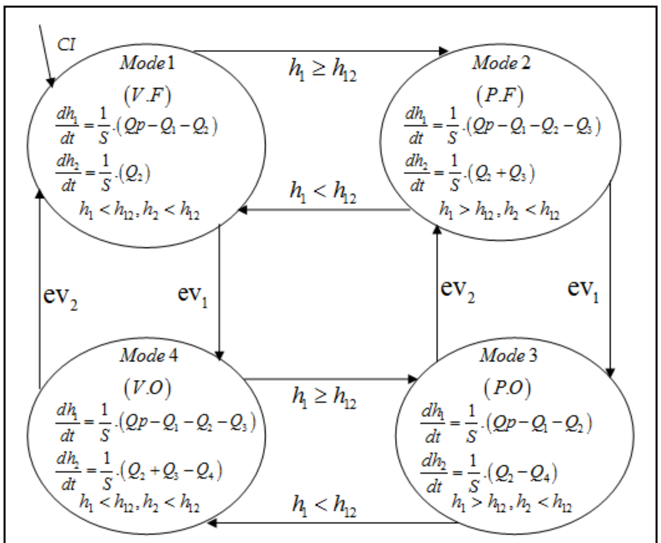

Fig. 8. Hybrid Automaton In Normal Operation.

Using these behaviors that are developed based on considered specifications, we can represent each behavior of this system by mode i, knowledge that switching from one mode to another is controlled by inequality constraints that depends on h_1, ev_1 or ev_2. Figure 8 illustrates the hybrid automaton in normal operation. Figure 9(a) and (b) illustrates the levels h_1 and h_2 and the evolution of modes respectively.

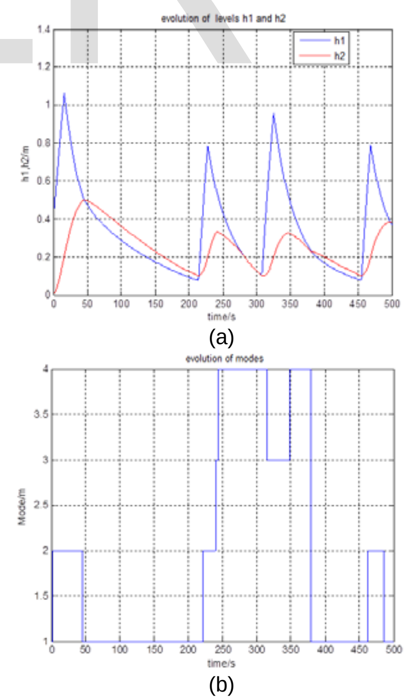

Fig. 9. (a) level of h_1 and h_2. (b) evolution of Modes

4.2 Modeling of the Monitoring System

We based this in part on the algorithm that we proposed previously, in which we have modeled the system behavior during normal and faulty operation by SHA. This algorithm allows us to extract the guards of input and output transitions of faulty operation. Therefore, we have been able to achieve monitoring system, described below.

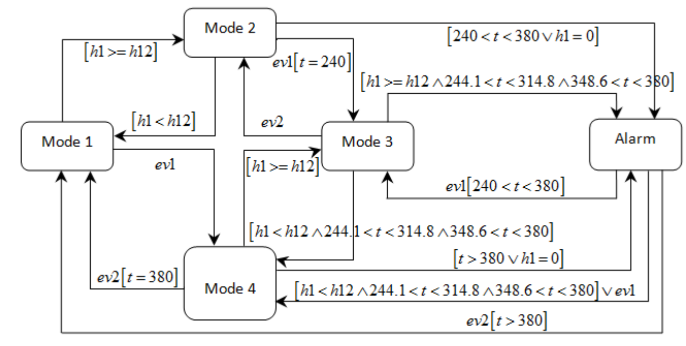

Fig. 10. Monitoring System Model.

As part of the formalization of our monitoring approach, we consider intermittent and permanent faults vis-à-vis which we analyzed behavior of our monitoring system. The table below shows the faults considered.

4.3 Results

Fault 1: The fault d_1 is characterized by blocking of the valve V_4 in opening mode until time t=385s, opening time exceeds the specified time $t \in [240 \;\; 380]$.

TABLE 2
CONSIDERED FAULTS IN THE SYSTEM

Faults	Type	Description
d_1	Intermittent	Blocking in opening of valve V_4.
d_2	Intermittent	Blocking in closing of valve V_4
d_3	Permanent	Final judgment of pump P_1

Figure 11 shows that an intermittent fault has been announced by the monitoring system (illustrated by the red circle in this figure). The alarm turns on quickly when t=380s for a few moments to tell us that the system has violated specific time (Figure 11(b)).

Fault 2: The fault d_2 is characterized by blocking of the valve V_4 in closing mode until time t=243s. In this case, an intermittent fault has been again declared by the developed monitoring system which it can be observed in Figure 12. The alarm turns on quickly when t=240s for a few moments (Figure 12(b)) to tell us that the system has violated specific time.

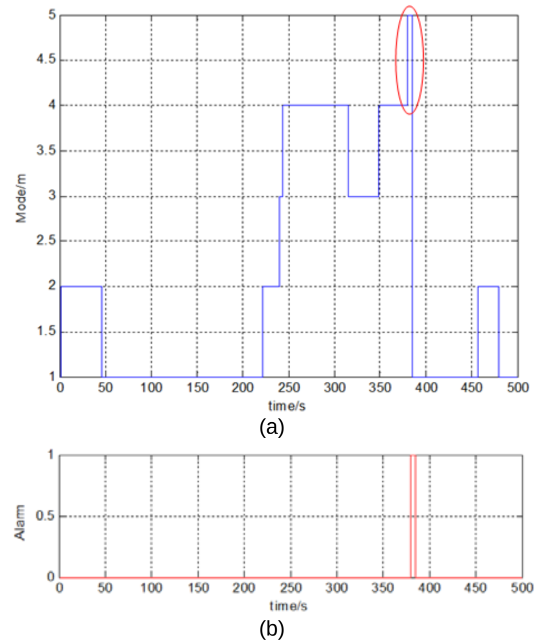

Fig. 11. (a) SHA Behavior In The Presence Of d_2 .
(b) Alarm Behavior

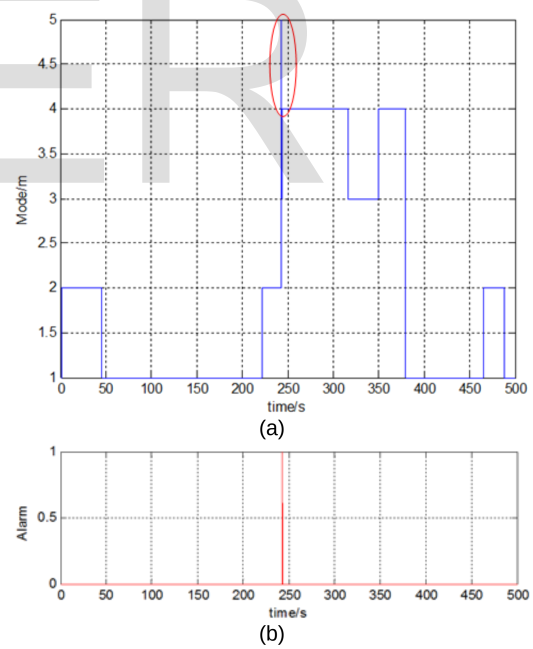

Fig. 12. (a) level of h_1 and h_2. (b) evolution of Modes

Remark 3 The two faults d_1 and d_2 do not affect the continuous part of our system, because the system is provided with a control of pump to keep level h_2 between h_{2min} and h_{2max}.

Fault 3: The fault d_3 occurs at Final judgment of pump at time t=243s. This type of fault affecting the continuous part of the system (Figure 13(a)) As it can be observed in Figure 13, a permanent fault has been correctly detected by the monitoring system (shown by the red circle in this figure). the alarm turns on quickly when t=243s for an unlimited delay (Figure 13(c)) to tell us that system is broken down, Which requires us to repair or to replace the pump.

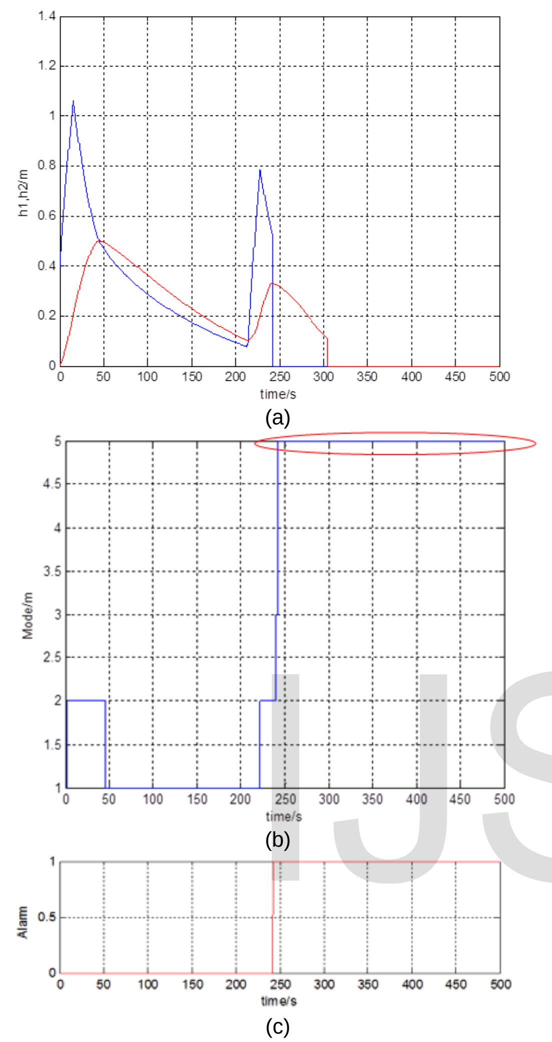

Fig. 13. (a) Continuous Evolution in The Presence of d_3. (b) SHA Behavior. (c) Alarm Behavior

5 Conclusion

In this paper we proposed a monitoring system for HDS whose model is a stopwatch hybrid automaton. It takes into account the time aspect and the dynamic changes that may appear during the process execution; this change may be definitive or spontaneous. The guards of automaton represent the crossing conditions, these conditions are timed with a timer that calculates elapsed time of each execution cycle. Authorized the behavior system (normal) is controlled by variables whose constraints are expressed by inequalities defining the acceptable space of system evolution.

The current work is devoted to developing a stopwatch hybrid automaton structure for a monitoring system able to detect, identify and locate the intermittent and/or permanent faults quickly and at the same time, by combining monitoring approach based on the model presented in this paper with the methods which are cited in section III.

References

[1] E. Y. Chow, and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems", *IEEE Transactions on Automatic Control*, vol. 29(7), pp. 602–615, July 1984.

[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis "Failure diagnosis using discrete event models", *IEEE Transactions on Automatic Control*, vol. 4(2), pp. 105–124, 1996.

[3] S. Tripakis "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, pp. 205-224, 2002.

[4] G. Zwingelstein "Diagnostic des défaillances. Traité des nouvelles technologies", série Diagnostic et Maintenance, Hérmes, 1995.

[5] S. H. Zad, R. H. Kwong and W. M. Wonham "Fault diagnosis in discrete-events systems: Incorporating timing information", *IEEE transaction on automatic control*, vol. 50(7) , 2005.

[6] T. El Mezyani, "Méthodologie de surveillance des systèmes dynamiques hybrides", thèse de doctorat, Université des Sciences et Technologies de Lille, 2005.

[7] A. Allaham. "Surveillance des systèmes à événements discrets commandés: Conception et implémentation en utilisant l'automate programmable industriel", *thèse de doctorat, L'université Joseph Fourier, Grenoble*, 2008.

[8] F. Lin, et W. M. Wonham. "Diagnosability of discrete event systems and its applications", *Discrete Event Dynamic Systems*, 1994, vol. 4, no 2, p. 197–212.

[9] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen et D. Teneketzis. "Diagnosability of discrete event systems", *IEEE Transactions on Automatic Control*, vol. 40, n°9, pp. 1555–1575, 1995.

[10] S. H. Zad, R. H. Kwong and W. M. Wonham. "Fault diagnosis in finite-state automata and timed discrete-event systems", dans *Proceedings of the 38th IEEE Conference on Decision and Control* 1999.

[11] M. F. Karoui, "Surveillance des processus dynamiques événementiels", Thèse de doctorat de l'université de Grenoble, 2011.

[12] H. Derbel, "Diagnostic à base de modèles des systèmes temporises et d'une sous-classe de systèmes dynamiques hybrides". Thèse de doctorat de l'université joseph fourrier, Grenoble, 2009.

[13] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. h. Ho, X. Nicollin, A.Olivero, J. Sifakis and S. Yovine. "The algorithmic analysis of hybrid systems", *Theoretical Computer Science*, vol. 138, pp. 3–34, 1995.

[14] V. Cocquempot, T. El Mezyani and M. Staroswiecki, "Fault Detection and Isolation for Hybrid Systems using Structured Parity Residuals", *ASCC'2004; Asian Control Conference*, Juillet 2004.

[15] L. Ghomri, "Synthèse de contrôleur de systèmes hybrides à flux continu par réseaux de Petri hybrides", Thèse de doctorat de l'université ABOU-BEKR BELKAÏD – Tlemcen, mars 2012.

[16] Pettersson, S., & Lennartson, B. Hybrid modelling focused on hybrid Petri nets. In *Proceedings of the 2nd EuropeanWorkshop on Realtime and Hybrid Systems* (pp. 303–309). Grenoble, France, (1995).