International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 6, December 2015, pp. 1396~1406
ISSN: 2088-8708

Using Hybrid Automata for Diagnosis of Hybrid Dynamical Systems

Lotfi Mhamdi*, Lobna Belkacem*, Hedi Dhouibi*, Zineb Simeu Abazi**
* LARATSI, National Engineering School of Monastir, Tunisia
** G-SCOP, Grenoble- IPN/UJF- Grenoble 1, CNRS, France

Article Info

Article history:

Received Apr 3, 2015
Revised Jul 15, 2015
Accepted Aug 1, 2015

Keyword:

Diagnosis
Hybrid dynamical system
Hybrid automata

ABSTRACT

Physical systems can fail. For this reason the problem of identifying and reacting to faults has received a large attention in the control and computer science communities. In this paper we study the fault diagnosis problem and modeling of Hybrid Dynamical Systems (HDS). Generally speaking, HDS is a system mixing continuous and discrete behaviors that cannot be faithfully modeled neither by using formalism with continuous dynamics only nor by a formalism including only discrete dynamics. We use the well known framework of hybrid automata for modeling hybrid systems, because they combine the continous and discretes parts on the same structure. Hybrid automaton is a states-transitions graph, whose dynamic evolution is represented by discretes and continous steps alternations, also, continous evolution happens in the automaton apexes, while discrete evolution is realized by transitions crossing (arcs) of the graph. Their simulation presents many problems mainly the synchronisation between the two models. Stateflow, used to describe the discrete model, is co-ordinated with Matlab, used to describe the continuous model. This article is a description of a case study, which is a two tanks system.

Corresponding Author:

Lotfi Mhamdi,
LARATSI, National Engineering School of Monastir,
University of Monastir,
Monastir, Tunisia.
Email: lotfienim@yahoo.fr

1. INTRODUCTION

In modern complex systems continuous and discrete dynamics interact. This is the case of wide manufacturing plants, agents systems, robotics and physical plants.

This kind of systems, called hybrid in their behaviour, needs a specific formalism to be analysed. In order to model and specify hybrid systems in a formal way, the notion of hybrid automata has been introduced [1], [2].

Intuitively, a hybrid automaton is a “finite-state automaton” with continuous variables that evolve according to dynamics characterizing each discrete state. In the last years, a wide spectrum of modeling formalism [3] and algorithmic techniques has been studied in the control and computer science communities to solve the problems of simulation, verification and control synthesis for hybrid systems [4], [5].

The control algorithms are generally developed considering that the system works in normal situation, i.e. is not faulty. Unfortunately, when failures occur, these algorithms become inefficient and even dangerous for the system itself or its environment. In order to reach higher performances and more rigorous security specifications, a Failure Detection and Isolation (FDI) [6] system has to be implemented.

In this paper we concentrate our attention to the problem of fault detection and isolation for hybrid systems.

The literature in that field is abundant and different solutions have been proposed for example the approach of [7], this method of model-based FDI algorithms is to compare the expected behavior of the system, given by a model which is modeled by hybrid automata, with its actual behavior, known through the online observations. ARR-based residuals are indicators of behaviors and thus may be used for FDI: they are equal to zero in normal (no-fault) situations and different from zero when the faults they are sensitive to, occur. A structured set of residual, i.e. a set of residuals that are not sensitive to the same subsets of faults, may be used to isolate the faults.

In the context of our work, we consider a diagnostic system based on control of the execution time of the tasks during the operation of the system; given that the system operation corresponds to the execution of all tasks of the process for well-defined time intervals. This method is based on a general modeling approach using hybrid automata. This temporal model is used to fault detection and isolation the faster possible and who diagnoses more precisely as possible by finding fault system components will be presented by using Stateflow controller. Indeed, if the diagnosis is fast and the failed component is identified, maintenance operations can be made more quickly.

The paper is organized as follows. Section 2 describes the hybrid dynamical system model in normal operating condition by hybrid automata. In Section 3, we explain the objective of our diagnostic approach based on hybrid model. The two tanks system is considered in section 4 to show the effectiveness of our diagnosis approach. At the end, a conclusion is presented with some perspectives.

2. MODELLING OF HYBRID SYSTEM

Two classes of hybrid model are distinguished [8]. The first class, known as integrated formalism, extends one of the models (discrete or continuous) in order to specify and describe the system. The second class of models co-ordinate the discrete model and the continuous one; this is the approach that we have taken. This choice is due to the fact that using a model for each component retains the specification potential of each domain. Continuous and discrete aspects correspond to two different worlds presenting two different views of a system.

In this section, hybrid automata model belonging to the second class is described. The construction of the diagnosis is based on this normal operating model which represents the system in normal situations, when no fault is present. Procedures based on this model are a priori able only to detect faults. When information is provided about which part of the normal operation model is unverified in presence of faults, isolation (location) of the faulty component becomes possible.

2.1. Hybrid Automata

Hybrid automata [9], [10] can be seen as an extension of timed automata with more general dynamics. A clock *x* is a continuous variable with time derivative equal to 1, that is $\dot{x} = 1$. In a hybrid automaton, the continuous variables *x* can evolve according to some more general differential equations, for example $\dot{x} = f(x(t))$. This allows hybrid automata to capture not only the evolution of time but also the evolution of a wide range of physical entities. The discrete dynamics of hybrid automata can also be more complex and described with more general constraints. In the following, we present a commonly used version of hybrid automata. Different forms of constraints result in different variants of this model. A hybrid automaton A consists of a finite set Q of discrete states and a set of *n* continuous variables evolving in a continuous state space $X \subseteq \mathbb{R}^n$. In each discrete state $q \in Q$, the evolution of the continuous variables are governed by a differential equation: $\dot{x} = f_q(x(t))$. The invariant of a discrete state q is defined as a subset I_q of X. The conditions for switching between discrete states are specified by a set of guards such that for each discrete transition T_q. A state (q, x) of A can change in two ways as follows:

1) by a continuous evolution, the continuous state x evolves according to the dynamics fq while the discrete state q remains constant;

2) by a discrete evolution, *x* satisfies the guard of an outgoing transition, the system changes discrete state by taking this transition. Let us consider the hybrid automata given in

Figure **1**.

This automata has three discrete states q_1, q_2 and q_3.

The continuous evolution in the states is represented respectively by $\dot{x}_1 = f_1(x), \dot{x}_2 = f_2(x)$ and $\dot{x}_3 = f_3(x)$.

The invariant in the state q_1, q_2 and q_3 are respectively $inv(q_1)$, $inv(q_2)$ and $inv(q_3)$.

The initial state of this system is represented by an input arc in the origin state q_1 , $init(q_1)$.

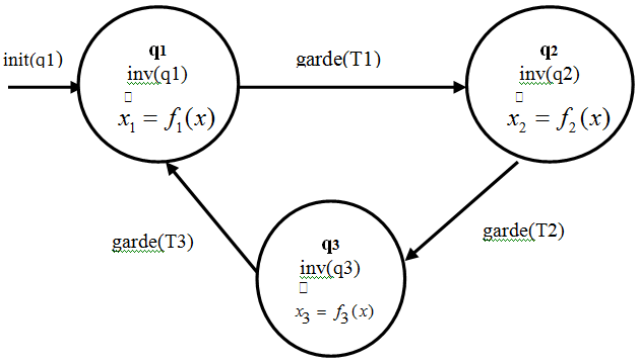

Figure 1. Hybrid Automata

Through the use of the hybrid automata one must create a normal operation model which represents the system in normal situations, when no fault is present. This model is obtained from identification of different possible states of the system, the evolution equations in each state and the necessary conditions for the transitions from one state to another. This dynamic model is neither more nor less a copy of the program control-command system to diagnose, to which is added time information, such as the duration of the different step of operation and date of occurrence events. Procedures based on this model are a priori able only to detect faults. When information is provided about which part of the normal operation model is unvaried in presence of faults, isolation (location) of the faulty component becomes possible.

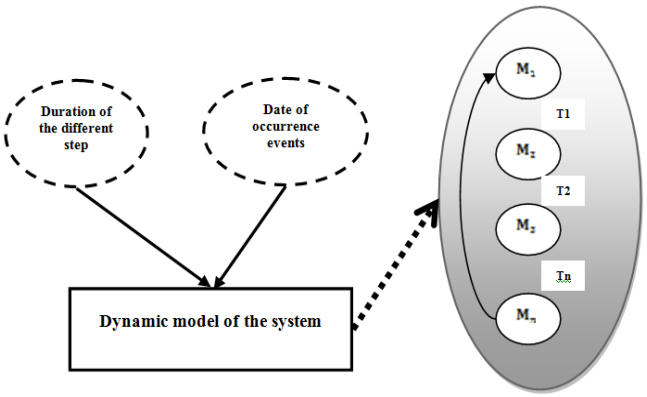

Figure 2. Representation of the dynamic mode l

Both continuous and discrete variables are necessary to describe the behavior of a hybrid system. The time evolution of the system results in a succession of modes. Each mode $(M_1, M_2 ... M_n)$ is characterized by a modality of the discrete state, a set of equality constraints (equations of state) and definition of domain admissibility (written by inequality constraints: invariant). A discrete transition T_n of a mode to another mode occurs when certain logical conditions are satisfied. These changes may be caused by discrete events that are generated by discrete actuators or sensors. The activity time of an associated mode state is specific to a given situation. Therefore we say that the system is in normal mode, if the mode activation time is noted in the interval I_m, if the activation time exceeds terminal $T_{\max}^{M_i}$ the system is considered faulty. It is represented by

Figure **3**. For each mode, we define two time values, as follows:

1) The minimum time necessary for correct execution of mode M_i , noted by $T_{\min}^{M_i}$

2) The maximum time tolerated for the execution of mode M_i , noted by $T_{\max}^{M_i}$

Thus we define the normal operating intervals noted, $I_m = \left[T_{\min}^{M_i}, T_{\max}^{M_i}\right]$ and that the faulty mode noted: $\left]T_{\max}^{M_i}, +\infty\right[$.

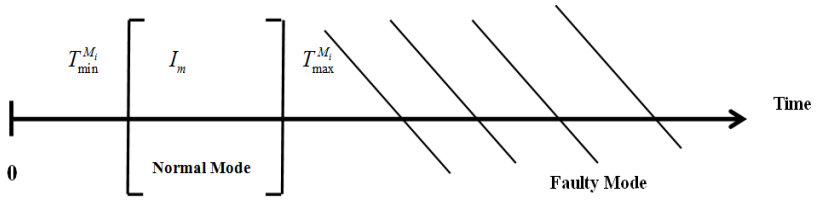

Figure 3. Activation time and mode of operation

The utility used for dynamic models is the hybrid automata, we shall also see that this tool has the disadvantage of increasing the size of the models considerably with the complexity of the system studied (particularly industrial systems). Thus, we must first answer the question "Is what all modeled trajectories are actually achievable in the system » Answer that question back to the reachability analysis of states in the graph which leads to reduced model of hybrid automata called attainable automaton which we modeled the possible evolutions of the system for a given initial condition.

3. OBJECTIVE OF OUR DIAGNOSIS APPROACH BASED ON HYBRID MODEL

In this paper we base ourselves on hybrid models, which propose the compilation of a diagnoser from a hybrid automata model of the system. In Figure 4, we illustrate the global schema of diagnoser construction.

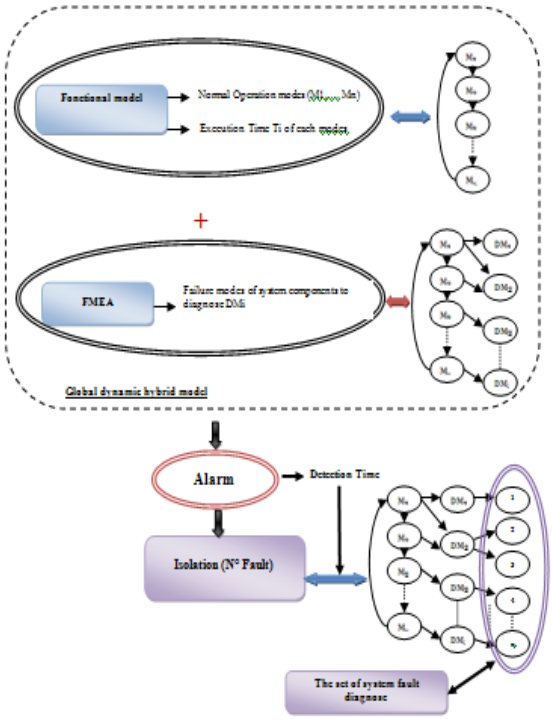

Figure 4. Etape of construction Diagnoser

According to the model of the system to diagnose, the event behaviors, temporal and differential equations of the process are identified. Thus, the model of the system is supposed to be "complete" in the sense it has present the normal and failing behavior of the process by the tool FMEA (Failure Modes and Effects Analysis).

And therefore we want to build the diagnosis of this system, in the form of a hybrid model. The role of the diagnosis is to infer the existence non- observable faults based on observable events and the time elapsed between these events.

A system is said to be faulty when the actual behavior is not consistent with one nominal trajectory. Even if faults are related to physical components (actuators, sensors, and system components) they may be classified with respect to their effects: faults may either affect the continuous-time evolution in a given mode (so-called continuous faults) or may affect the discrete trajectory (so-called discrete faults) [11].

3.1. Continuous Faults

Continuous faults are related to a given mode. They may be of two types:

1) Fault that corrupts the equality constraints.
2) Fault that corrupts the inequality constraints.

3.2. Discrete Faults

These faults perturb the discrete evolution. Three kinds of faults may be considered:

1) The system is moving from one mode i to another one which is not referred as a possible successor if the system works normally, that is to say a successor which does not belong to the prediction graph of level 1 associated with mode i.
2) The system is moving from one mode i to a successor that belong to the prediction graph associated with mode i but the transition condition is not verified.
3) The system is staying in a mode even though a spontaneous or forced switching condition is validated.

To better understand the different phases of diagnosis (construction of the dynamic model, detection phase, location) we will describe in more detail these ideas through a hydraulic system with two tanks.

4. APPLICATION: TWO TANKS SYSTEM

4.1. Description of the System

The two tanks system depicted in Figure 5 is considered to illustrate the diagnosis methodology. The system consists of:

1) 2 tanks R1 and R2, whose sections are equal S1 = S2 = 0.0154 m^2. These tanks are linked by a lower pipe C_2 and an upper pipe C_3. The flow through pipe C_2 can be interrupted with a switching valve V_2,
2) One pump P that delivers a liquid flow Q_p that fall into tank R1,
3) 4 switching valves V_1, V_2, V_3 and V_4 allow to control the flows Q_1, Q_2, Q_3 and Q_4,
4) Two level sensors Sh1et Sh2 (respectively to measure the level in both tanks R1 and R2),
5) An overflow sensor Dh1 in the reservoir R1.

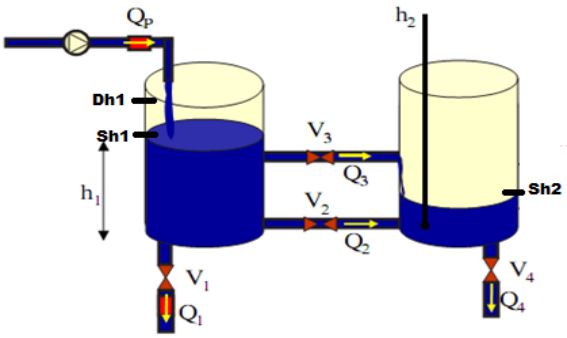

Figure 5. Two tanks system

The pump is controlled in on–off-control so as to maintain h_2 in a fixed interval. The logic of the pump is as follows:

1) The pump is initially turned on,

2) She stopped when $h_2 \geq 0.2m$,

3) It is switched on when $h_2 \leq 0.1m$

The pump flow is measured, the state of the pump (on or off) is not considered thereafter. When the pump is stopped, zero flow ($Q_p = 0$), when operating $Q_p = Q_0 = 0.001$ m^3/s.

The system may be modeled by considering 5 discrete states. The first one is the state of pipe C_3 that may take the two modalities: Empty (E) or Full (F). The 4 other discrete states are the states of valve V_1, valve V_2, V_3 and valve V_4 that may take the modalities Opened (O) or Closed (C). As a consequence 32 modes allows to represent all possible normal situations. Each mode is characterized by a modality of the discrete state vector (V_3, V_1, V_2, V_3, and V_4), a set of continuous state equations and inequality constraints. The four valves V_1, V_2, V_3 and V_4 are controlled manually. They may be opened or closed by the operator at any time. We consider that the system is used in a given exploitation mode in which V_1 and V_2 are always opened. Only pipe C_3 and valve V_4 are operated. The two corresponding actions (open and close) correspond in the following to events e_1 and e_2. The events e_1 and e_2 (respectively the time of opening and closing the valve V_4) are controlled; e_1 occurs at time t = 240 s, while e_2 occurs at time t = 380 s.

The flows Q_1, Q_2, Q_3 and Q_4 are given by:

1) $Q_1 = \alpha\sqrt{h_1}$

2) $Q_2 = \alpha\, sign(h_1 - h_2)\sqrt{|h_1 - h_2|}$

3) $Q_3 = \alpha\, sign(h_1 - 0.5)\sqrt{|h_1 - 0.5|}$

4) $Q_4 = \alpha\sqrt{h_2}$

With: $\alpha = A\sqrt{2g}$; où: $A = 3.6\times10^{-5} m^2$ and $g = 9.81 m/s^2$ is gravity.

Q_1 and Q_4: the outflows respectively R1 and R2 tanks.

Q_2 and Q_3: the outflows from the reservoir R1 to R2 through pipes C_2 and C_3, respectively.

The hybrid automaton that represents the system under normal conditions in this exploitation mode is part of the complete automaton and is given by

Figure **6**.

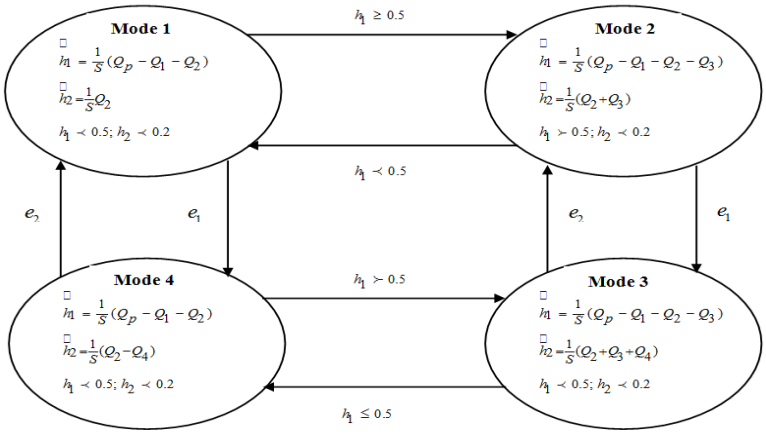

Figure 6. Hybrid automata: normal behavior

From dynamic model construction we go to explain the different phases of the diagnostic with tool Matlab / Simulink/Stateflow.

4.2. Construction of the Diagnoser with Stateflow

Control of Hybrid Power Plant stateflow is a tool integrated in the MATLAB environment and used for the development and the simulation of complex reactive systems. It uses a variant of the finite state machine. Specifically, it uses the hybrid State charts formalism. It provides a block that can be included in a Simulink model. Additionally, it enables the representation of hierarchy, parallelism and

history. Hierarchy enables the organization of complex systems by defining a parent/offspring object structure.

4.2.1. Normal Behavior

Simulation of the TWO TANKS SYSTEM with State flow is depicted in Figure **7**.

1) Mode 1: you must complete reservoir R1 to the level h_1 attained 0.5.
2) Mode 2: the level h_1 reached 0.5, the two pipes C_2 and C_3 are open, you must complete reservoir R2 to the level h_2 attained 0.5.
3) Mode3: Where the two levels tanks exceed limits, the valve V_4 is opened to the time e_1.
4) Mode 4: we must empty the two tanks to level h_1<0.5.

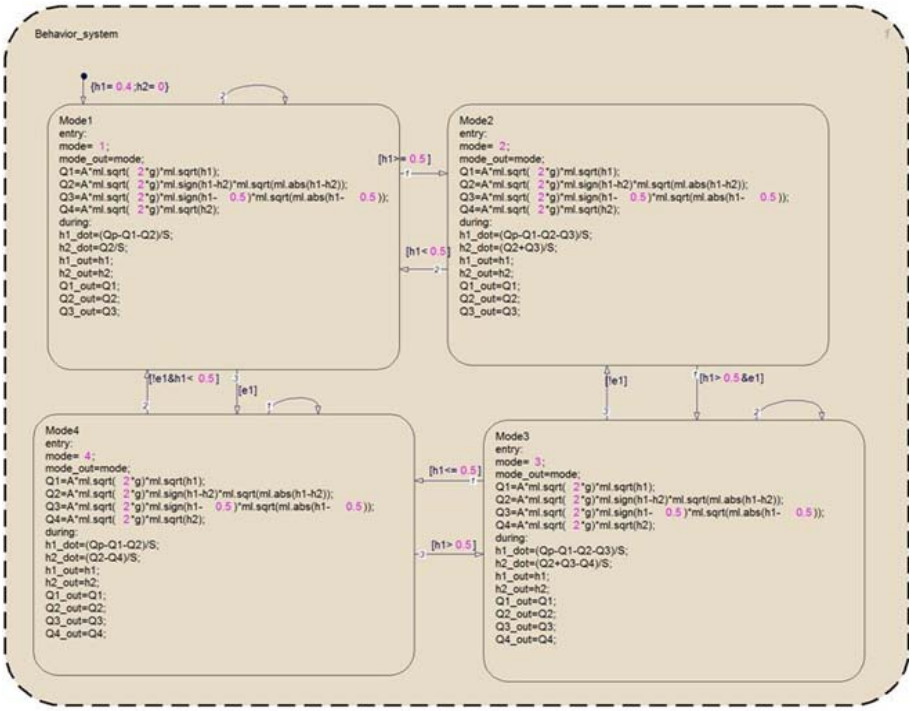

Figure 7. Modeling of the process with state flow

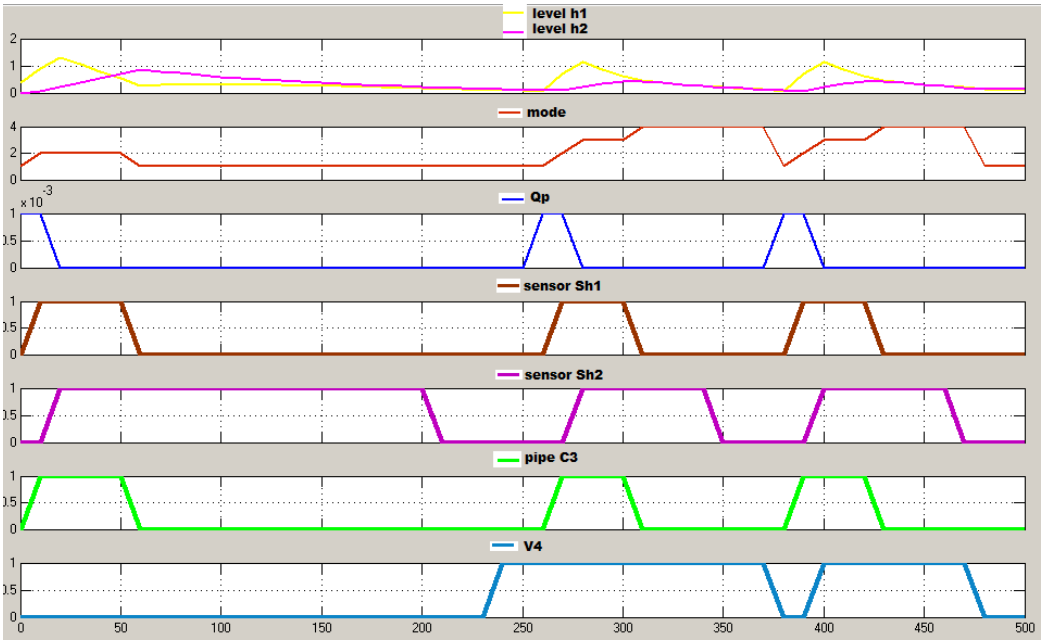

Figure 8. Normal behavior

The construction of the diagnosis is based on the temporal knowledge of the process; we need to know the process duration such as the opening time of the valves or the time of sensor status change. Figure 8 illustrates the nominal behavior of the instrumentation process of an operating cycle.

From figure above, the transition times are defined for each phase of the normal process.

Table 1. Identification of temporal process

Actions	Time in sec	Interval of time
Opening the pump Qp.	0	[0 , 20]
Closing the pump Qp.	20	
Opening the valve V_4	240	[240 , 380]
Closing the valve V_4	380	
Activating the sensor Sh1	7.7	[7.7 , 60]
Deactivating the sensor Sh1	60	
Activating the sensor Sh2	20	[20 , 180]
Deactivating the sensor Sh2	180	

4.2.2. Considered Failures

In order to illustrate the possible fault cases stated at section 3.1 and 3.2 (continuous and discrete faults), we consider the following 2 particular situations:

1) The faults that perturb the state equations (continuous faults):
   a) Fault of sensor Sh1 that measures h_1,
   b) Fault of sensor Sh2 that measures h_2.
2) The faults that perturb the passage between different modes (discrete faults):
   a) Event e_1 is controlled. It occurs time t = 240s. If this event has no effect on the discrete state evolution (valve V_4 stays blocked opened), the system will stay in mode 2 (The passage mode 2 to mode 3 is disabled),

b) Event e_2 is controlled. It occurs at time t = 380s. If this event has no effect on the discrete state evolution (valve V_4 stays blocked closed), the system will stay in mode 4 (The passage mode 4 to mode 1 is disabled),
c) In fact, if pipe P3 is entirely clogged, the dynamics of the state variables will continue to correspond to the continuous state equations of mode 1 even if the level h_1 becomes larger than 0,5m (the system rest in mode 1).

4.2.3. Fault Diagnosis Using Stateflow

In order to detect and locate a fault on the studied process, a system for injecting random defects on the process instrumentation was created.

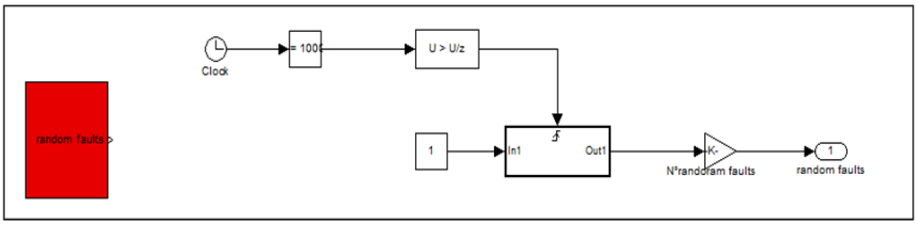

Figure 9. Injection block random faults

5. RESULT APPLICATION

In this work we are interested only in detecting the overflow of the reservoir R2 (given by h_2) and we consider one faults related to the reservoir: Sh2_stuck close (F_{Sh2} : Does not detect the lower level (always set to 1)). This behavior is represented by the faulty model (Figure 10).

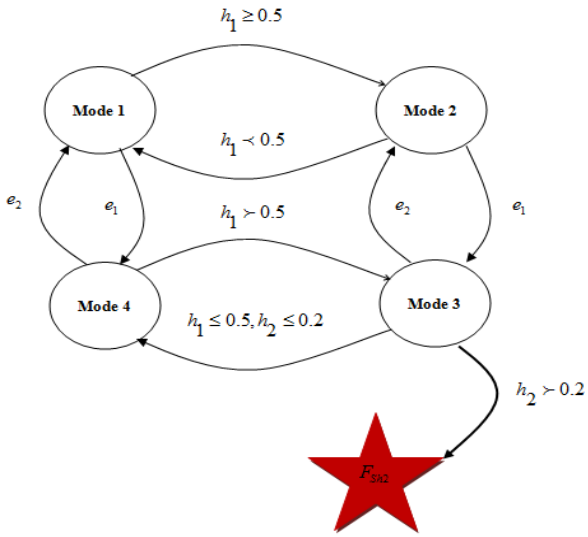

Figure 10. Model of the faulty system

Our objective is to detect and identify the faults occurring in the process. That leads to determining the way of locating a fault, and to determining the time of its occurrence. Figure 11 describes the variations of the level in the two Reservoirs.

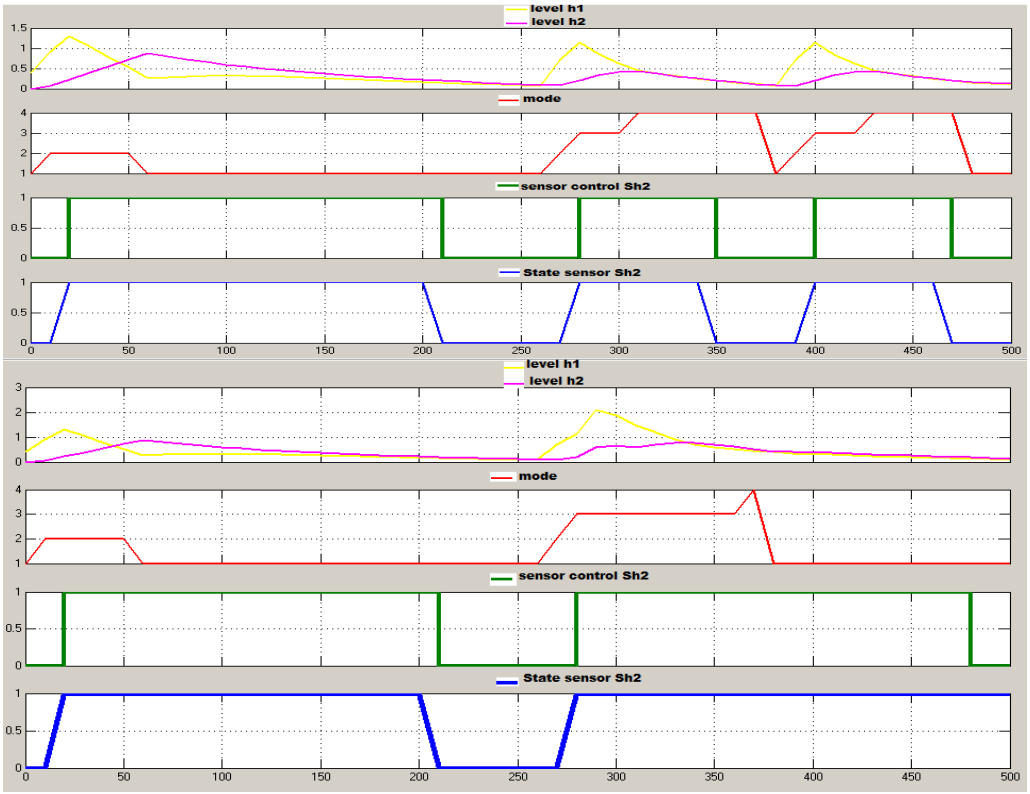

Figure 11. Diagnosis for fault Sh2_STUCK CLOSE

To better understand the principle of the diagnoser, Figure 11 allows us to compare normal operation (part above) of the process with a state of faulty operation (part below).

In the figure below, despite the level sensor detects lower reservoir R2 level that is to say, pass to the zero state (green signal), it remains in state 1 (blue signal). This moment represents the occurrence of a failure. Since the sensor Sh2 remains in state 1 (h_2> 0.2); therefore there is disturbance on the evolution of the continuous part is due to the passage mode 3 to mode 4 is canceled.

5. CONCLUSION

In this paper we studied the fault-diagnosis problem for hybrid systems. We used the formalism of hybrid automata for modeling hybrid systems with faults and to define the notions of diagnosability and time abstract diagnosability. We focused our attention on time-abstract diagnosability and we defined a Fault Diagnosis on hybrid automata with faults for TWO TANKS SYSTEM. However, our approach can be integrated with the prognosis. This is to be capable both of responding to the occurrence of a fault, but also to be able to anticipate.

ACKNOWLEDGEMENTS

The authors would like to thank the comments provided by the anonymous reviewers and editor, which help the authors improve this paper significantly. We would also acknowledge the help from Zineb Simeu-Abazi, Institut National Polytechnique de Grenoble, INPG France.

REFERENCES

[1] 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 138*, pp. 3–34, 1995.

[2] O. Maler, Z. Manna, and A. Pnueli, "From Timed to Hybrid Systems", In J. W. de Bakker, C. Huizing,W. P. de Roever, and G. Rozenberg, editors, Real-Time: Theory in Practice, 600, Springer-Verlag, pp. 447–484, 1991.

[3] Saeed Ghasemi, Leili Mohammad Khanli, Mina Zolfi, Ghader Tahmasebpour, "Modeling and Simulation of NFC Logical Layer Peer-to-Peer Mode using CPN and TA", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 4, No. 2, pp. 162-168, 2014.

[4] Branicky M.S., "Studies in hybrid systems: Modeling, Analysis and Control", PhD thesis, MIT, Massachusetts, USA, 1995.

[5] Engell S., "Modelling and analysis of hybrid systems", *2nd IMACSMATHMOD Conference*, Vienne, Autriche, pp. 17031, 1997.
[6] B. Bellali, A. Hazzab, I. K. Bousserhane, and Dimitri lefebvre, "A Decoupled Parameters Estimators for in Non linear Systems Fault diagnosis by ANFIS", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 2, No. 2, pp. 166-174, 2012.
[7] V. Coquempot, T. El Mezyani, and M. Staroswiecki, "Hybrid Dynamical Systems Monitoring using Structured Analytical Redundancy Relations", Proceedings of 17ème IMACS World Congress, Paris, France, 2005.
[8] R. Champagnat, H. Pingaud, H. ALLA, C. Roubinet, and J. M. Flaus. "A Gas Storage Example as a Benchmark for Hybrid Modelling", *ADPM'98 conf. On Automation of Mixed Process: Dynamical systems*, Reims, France, 1998.
[9] R. Alur. "NATO-ASI 1998 Summer School on Verification of Digital and Hybrid Systems", *Timed automata*, 1998.
[10] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. 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.
[11] Cocquempot V., T. E. Mezyani, and M. Staroswieckiy, "Fault detection and isolation for hybrid systems using structured parity residuals", *dans Asian Control Conference, ASCC'04*, New Mexico, Vol. 2, pp. 1204–1212, 2004.

BIOGRAPHIES OF AUTHORS

Lotfi Mhamdi received his Engineer degree of maintenance and master at National School of Engineering - University of Center, Tunisia in 2004 and 2007 respectively. In 2014, he obtained his doctorate degree in Industrial automation: automatic and Industrial computing from Institut National Polytechnique de Grenoble, INPG France and National School of Engineering of Monastir, University of Center. He is currently Assistant professor of Electrical Engineering at University of Kairawan Tunisia. His research interests include Modeling, Intelligente Control and Monitoring and command Manufactory systems, he is former member in LARATSI - (Labortoired'Automatique ,Traitement de signal et Imagerie), Monastir, Tunisia.

Lobna Belkacem was born in Teboulba, Tunisia, in 1987. She graduated from the National School of Engineering of Monastir, University of Center. She obtained her Engineer degree of electrical and Master in 2012 and 2013, respectively. Actually,she is preparing her doctorate degree in automation. She is former member in LARATSI - (Labortoired'Automatique ,Traitement de signal et Imagerie), Monastir, Tunisia.

Hedi Dhouibi - received his Engineer degreeof maintenance and DEA at National School of Engineering - University of Center, Tunisia in 1997 and 1999 respectively. In 2005, he obtained his doctorate degree in Industrial automation: automatic and Industrial computing from University of the sciences and the technologies of Lille France.
He is currently Assistant professor of Electrical Engineering at University of Kairawan Tunisia. His research interests include Modeling, Intelligente Control and Monitoring and command Manufactory systems.

Dr Zineb SIMEU-ABAZI is Assistant Professor at the Polytech'Grenoble in the University Joseph Fourier where she teaches control processing, Automation and Industrial Engineering, dependability and Industrial Maintenance. She holds a Ph.D. in Computer Science and Automation at the Institut National Polytechnique de Grenoble, INPG France on 1987 and an « Habilitation à Diriger des Recherches" » HDR in 1998, from Grenoble University, France. She is particularly interested in the on line maintenance, diagnostic, recycling and performance evaluation fields. In relation to these topics she took scientific responsibility of French and International projects/groups on e-maintenance such as the CNRS MACOD working group (Modelling and Optimisation of Distributed vs. Collaborative Maintenance).She is a president of the scientific council of diag21 association.