Discrete Event Model-Based Approach for Fault Detection and Isolation of Manufacturing Systems

Alexandre Philippot, Moamar Sayed-Mouchaweh, Véronique Carré-Ménétrier

Discrete event model-based approach for Fault Detection and Isolation of manufacturing systems

Article · January 2009

3 authors:

Alexandre Philippot
Université de Reims Champagne-Ardenne

M. Sayed Mouchaweh
Institut Mines-Télécom Lille Douai

Véronique Carré-Ménétrier
Université de Reims Champagne-Ardenne

Discrete Event Model-Based Approach for Fault Detection and Isolation of Manufacturing Systems

A. Philippot, M. Sayed-Mouchaweh, V. Carré-Ménétrier

Université de Reims Champagne-Ardenne - Centre de Recherche en STIC (URCA-CReSTIC)
Moulin de la Housse B.P. 1039, 51687 REIMS Cedex 2, FRANCE,
({alexandre.philippot, moamar.sayed-mouchaweh, veronique.carre}@univ-reims.fr)

Abstract: This paper presents a discrete event model-based approach for Fault Detection and Isolation of manufacturing systems. This approach considers a system as a set of independent plant elements. Each plant element is composed of a set of interrelated Parts of Plant (PoPs) modeled by a Moore automaton. Each PoP model is only aware of its local behavior. The degraded and faulty behaviors are added to each PoP model in order to obtain extended PoP ones. An extrapolation of Gaussian learning is realized to obtain acceptable temporal intervals between the time occurrences of correlated events. Finally based on the PoP extended models and the links between them, a fault candidates' tree is established for each plant element. This candidates' tree corresponds to a local on-line fault event occurrence observer, called diagnoser. Thus, the diagnosis decision is distributed on each plant element. An application example is used to illustrate the approach.

Keywords: Discrete-Event Systems, Decentralised/Distributed Models, Diagnosis, Automata, Manufacturing Systems.

1. INTRODUCTION

In complex systems, such as manufacturing processes, unpredictable events and undesired reactions are the consequence of the occurrences of faults. The later reduces significantly the systems performances. Manufacturing systems can be represented as Discrete Event Systems (DES), i.e., dynamic systems with discrete state spaces and event-driven transitions (Cassandra et al. 1999). Consequently, Boolean information is only available and its observation alone does not often allow to detect the fault occurrences and to isolate the responsible elements.

Several approaches have been developed to solve the Fault Detection and Isolation (FDI) problem (Darkhovski et al. 2003, Hadjicostis 2005, Rozé et al. 2002). FDI has become a crucial issue for industrial process diagnosis leading to increase availability, reliability and production safety. FDI approaches can be divided generally into knowledge-based and model-based ones. The knowledge-based, or model-reasoning, approaches construct a model about the system behavior based on an initial human experience, e.g. expert systems, on a set of historical data, e.g. pattern recognition and signal processing methods (Duda 2001). Model-based FDI approaches compare a mathematical and/or graphical (automata, Petri nets, …) model of the normal and/or abnormal behaviors of the system with its real input/output data. These models observe the system by their events in order to infer the fault occurrences. Thus, they are called "diagnosers" (Lamperti et al. 2008, Rozé et al. 2002).

Different structures of model-based approaches for diagnosing DES exist. The first structure is the centralized one which requires a global model of the system function as well as a global diagnoser (Sampath 1995). Constructing a global model is often intractable because of the complexity and the large size of the manufacturing systems. An alternative of the centralized structures is the component-oriented model approaches. In these approaches, the system global model is described by a set of local models available through a library. This description is realized by decentralized or distributed ways. In decentralized approaches (Wang et al. 05), the diagnosis is performed based on a set of local diagnosers. However, a global model of the system is required to take into account the links between the interrelated components. In distributed approaches (Boel et al. 2004, Philippot et al. 2007, Cordier et al. 2007), no need to global model. Each local model is only aware of its own behavior. The links, or dependencies, among components are considered via components exchange local diagnoses using communication protocols (Genc et al. 2003) or merging strategies (Cordier et al. 2007).

This paper proposes a modular and distributed FDI model-based approach for plant faults detection and isolation of manufacturing systems. This approach divides the plant into a set of Parts of Plant (PoPs). A PoP can be an actuator or a sensor. They are modeled by discrete models which take into account the technology specifications used to produce them. Potential degraded and faulty behaviors of each PoP are added to its model in order to obtain an extended PoP model. Faulty behavior causes the production halt while the degraded one disturbs or reduces the optimal production performances. The PoP models are grouped into subsets of interrelated ones. Each subset of interrelated PoP models defines an independent plant element. Finally, the possible candidates for a faulty or degraded behavior of each plant element are established based on its own extended PoP models and the links between them.

The paper is structured as follows. In section 2, the proposed model-based FDI approach is presented. The system model is described by its PoP models included in a library. The later contains a set of local models of PoPs commonly used in discrete manufacturing systems. In section 3, a manufacturing system is used to illustrate the approach. The last section concludes the paper and presents future research directions.

2. MODEL-BASED FDI APPROACH

2.1 Part of Plant

Generally, plant is composed of pre-actuators, actuators and sensors (Fig. 1). We consider the plant as a set of independent plant elements PE_k, $k \in \{1,2,..,m\}$. Each plant element is composed of a set of actuators and sensors. Each actuator or sensor is defined as a Part of Plant (PoP). A PoP can react to failure events by changing its state.

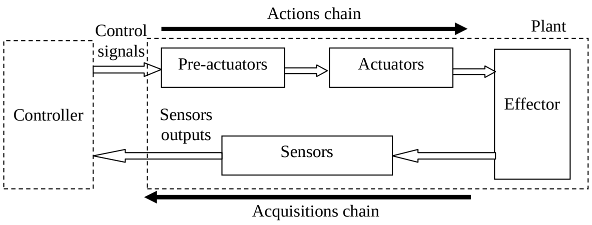


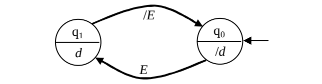
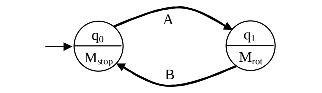
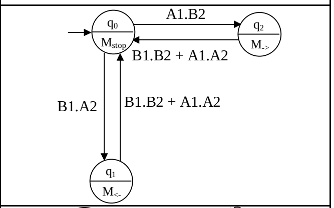
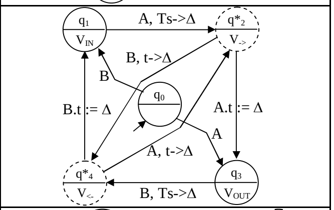
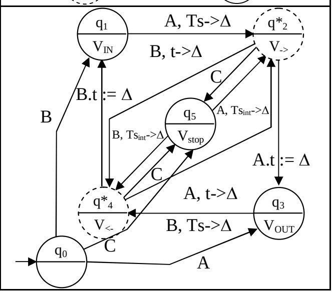
Fig. 1. Manufacturing system general structure.

We establish detailed PoP models which take into account the technological specifications (Deluche 1998) used to produce them. The goal is to obtain realistic models and to avoid combinatory explosion of DES modeling (Sampath 1995, Wonham et al. 1987). Each model is represented by a Moore automaton (Cassandra et al. 1999). The later is a Finite State Machine (FSM) in which the outputs are determined by the current state. Each PoP local model receives failure and internal events and emits its internal events and state outputs to the other interrelated PoPs.

A PoP model is represented by timed Moore automaton defined as a 11-tuple $G_{PoP} = (Q, q_0, \Sigma, \Lambda, T, O, Q^*, T_s, \Delta, t, \Psi)$ where:

- Q is a finite set of static states ;
- q_0 is the initial state belonging to Q ;
- Σ is a finite set of input events (input alphabet) ;
- Λ is a finite set of output events (output alphabet) ;
- ($T : Q \times \Sigma \rightarrow Q$) is a transition function mapping a state and the input alphabet to the next state ;
- ($O : Q \rightarrow \Lambda$) is an output function mapping each state to the output alphabet ;
- Q^* is a finite set of dynamic states (represented in dotted line in Table 1) ;
- (T_s) is a Time of stroke required for the displacement between two positions (transition between two states) ;
- Δ is a time variable of temporary allocation ;
- t is a local clock measuring the elapsed time between 2 events ;
- $\Psi = \{->, :=\}$ is a finite set of operands. (->) and (:=) refer, respectively, to allocation and equality test operations.

Table 1. Library of commonly used Part of Plant models in discrete manufacturing systems

	Part of Plant	Model
Sensors	Digital sensors with a binary response	
Actuators	Motor with 1 sense of rotation	
	Motor with 2 sense of rotation	
	Simple Acting Cylinder (SAC) and Double Acting Cylinder (DAC) with 2 positions	
	DAC with n positions (n>2 and with 5/3-way valve or equivalent intermediate position C)	

The construction steps of PoP models are described in (Philippot et al. 2008) and can be resumed in Table 1. As an example, let consider the model GDAC of the Double Acting Cylinder (DAC) with 2 positions. The model GDAC evolves from its initial state q0 towards the states q_1/VIN or q_3/VOUT according, respectively, to the activation of the control signals B = In or A = Out. The states q_1/VIN and q_3/VOUT represent, respectively, the piston rod in home return and in fully extended positions indicated, respectively, by the output events VIN and VOUT. If the model is located in the state q_1, the activation of the control signal A = Out leads the piston rod to move forward. This piston rod movement is represented by the dynamic state q^*_2/V->. The output event V-> indicates that the piston rod is in movement towards its fully extended position. The time required to reach this position, Ts, is assigned to the time variable Δ. In the same time, a local clock t is initiated to calculate the spent time during the forward movement. At this dynamic state, two cases can arise. In the first case, the value of t becomes equal to the one allocated to Δ. This means that the actuator has reached its fully extended position. Therefore, GDAC reaches the state q3 with the output event Vout. In the second case, the control signal B = In is activated. This activation forces the piston rod to stop moving forward in order to inverse its movement towards its home position. Thus, GDAC evolves to the dynamic state q*4 with the output event V<- indicating that the piston rod is in inversed movement. In this case, the present spent time t is assigned to Δ. Then, the local clock is initiated again to calculate the elapsed time in the inverse movement. When this time becomes equal to the one allocated to Δ, the piston reaches its home position indicated by the state q1/VIN. The same reasoning can be followed for the other states.

2.2 Abnormal or faulty states

We adopt the hypothesis that each behavior which does not correspond to a normal one is considered as abnormal one. Thus, starting from normal states of each PoP model, it is possible to determine the abnormal (degraded or faulty) states. An abnormal state is reached due to the occurrence of a failure event, which is unobservable event, at a normal state. The abnormal states are represented by a square in the extended PoP models. For the example of the DAC with 2 positions, these abnormal states indicate the following faults:

- The DAC is stuck or blocked (B) ;
- The DAC reacts too slowly to the control signals in comparison with its normal behavior (D).

However, these abnormal states require the determination of the intervals defining the acceptable time displacement of the DAC. To determine these intervals, we have established a learning phase about the system's normal and abnormal behaviors. The goal of this learning is to obtain realistic time response intervals related to the system dynamic and to the PoP technology. These intervals are obtained by an extrapolation of Gaussian learning defining the probability, chance, of the occurrence of an event in this interval. The temporal constraints between the time occurrences of correlated events can then be represented by a template (Holloway et al. 1994) or a chronicle (Cordier et al. 2000).

We use the Balemi's interpretation. In (Balemi et al. 1993), the authors define controllable events $\Sigma_c \subseteq \Sigma$ as controller's outputs sent to actuators, and uncontrollable events $\Sigma_{uc} \subseteq \Sigma$ as the controller's inputs coming from sensors. $\Sigma_o = \Sigma_c \cup \Sigma_{uc}$ is the set of observable events and is included in Σ. It considers that a change of a variable α from 0 to 1, or from 1 to 0 produces events characterized by either rising, ↑α, or falling edges, ↓α. Consequently, the observation of the system's, actuators, reactivity is achieved by these sensors events, which are considered as uncontrollable events. The control events, produced by the activation of control signals, are considered as controllable events. The use of rising and falling edges facilitates the detection of permanent and intermittent faults.

2.3 Plant Elements

The plant of a system is composed by *n* Parts of Plant: PoP_i, i ∈ {1, 2… *n*}. Each local PoP_i is modelled by 11-tuple $G_{PoPi} = (Q_i, q_{0i}, \Sigma_i, \Lambda_i, T_i, O_i, Q_i^*, T_{si}, \Delta_i, t_i, \Psi_i)$ as defined section 2.1. A plant element PE_k is said to be "independent" of another one PE_h if their models do not have common inputs and outputs events. Consequently, each independent plant element contains an independent subset of PoPs from the other plant elements. If $\{PoP_{PE_k}\}$ denotes the set of PoPs constituting the Plant Element PE_k, then plant elements independency can be represented by:

$$\forall PoP_i \in \{PoP_{PE_k}\}, \forall PoP_j \in \{PoP_{PE_h}\} \Rightarrow \begin{cases} \Sigma_{PoPi} \cap \Lambda_{PoP_j} = \Phi \\ \Lambda_{PoPi} \cap \Sigma_{PoP_j} = \Phi \end{cases}$$

No need to construct a global model of a system if it is divided into a set of independent plant elements. The occurrence of a fault in a PoP belonging to a plant element will not be propagated to other plant elements. However, a plant element can possess relatively a big number of PoPs in order to be independent of the other plant elements. This depends of system and its structure. In the worst case, a system can be divided into one plant element if all the PoPs are interrelated.

2.4 Fault Plant Element Candidates

The candidates responsible of the occurrence of a fault in a PE can be determined based on its PoP models as well as on its temporal constraints represented by a set of templates or chronicles. The following two hypotheses are considered:

- Only one failure event responsible of a faulty or degraded behavior can occur at the same time ;
- Controller is supposed to be dependable and safety. Consequently, the controller cannot be responsible of any fault as the one of sending two opposable control signals.

Now, to determine all possible candidates responsible of an abnormal behavior in a Plant Element, a candidates' tree is constructed. It is based on a knowledge expert and on each control signal sent by the controller. A control signal entails only one Plant Element of the system to evolve. The response of this control signal is sensors' events leading to change their outputs. Consequently, the abnormal states can be calculated according to these outputs.

The number of candidates can be reduced using a progressive monitoring. The occurrence of new sensors events can lead to eliminate the improbable or inconsistent candidates with this new observation. The fault candidates' tree can correspond to an online diagnoser which returns one or several labels according to the observations. Furthermore, thanks to the results of Gaussian learning, it is possible to achieve a preference, i.e., more probable candidates for the occurrence of a fault. This preference is considered by the order of labels, i.e. candidates, in each state of the candidates' tree.

3. MANUFACTURING SYSTEM EXAMPLE

To illustrate the proposed approach, we use the example of pick and place station of the flexible manufacturing system platform *Cellflex* (http://meserp.free.fr/). This station realizes the import and the export of pieces by a gripper between two processes thanks to a pneumatic system on 3 axes (Fig. 2). The symbol “1” refers to *Z* axis displacement, “2” to *X* axis displacement, “3” to *Y* axis displacement and “4” to the pneumatic system gripper.

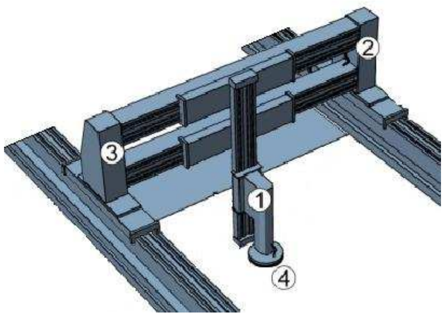


Fig. 2. Pick and place station

This station is composed of 4 actuators piloted by 6 pre-actuators produced by different technologies. The information about the behavior of the station is provided by 9 sensors (Fig. 3). These models have been constructed using the software *Stateflow* of Matlab® in order to generate their behavior by simulation.

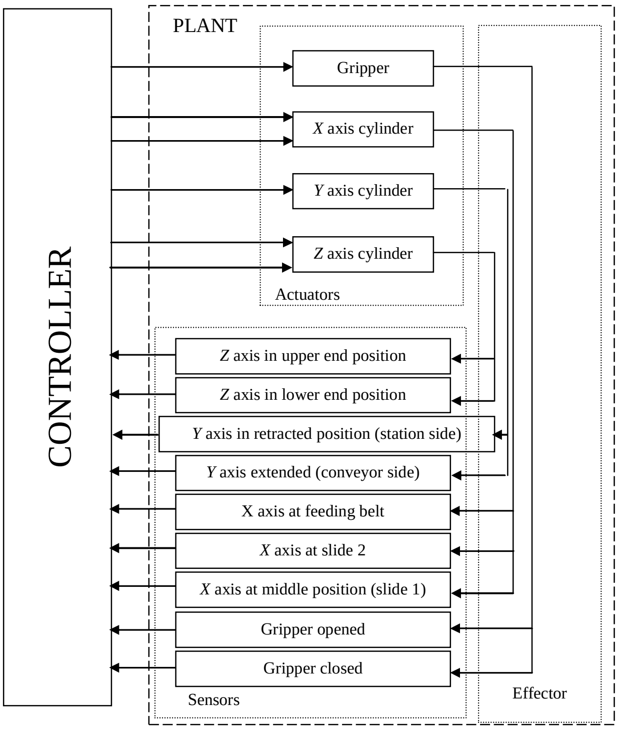


Fig. 3. PoP of pick and place station

3.1 PoP of *Y* axis PE

Based on Fig. 3, we can see the interactions between PoPs allowing gathering them in independent Plant Elements (PEs). For example, the *Y* axis is composed of 3 PoPs (one actuator and 2 sensors) which communicate among them. The actuator is a Double Acting Cylinder (DAC) where its retracted and extended positions are indicated by, respectively, the sensors y_R and y_E. These 3 interrelated PoPs constitute the “*Y* axis” PE. We keep this PE for the explanations hereafter (Fig. 4).

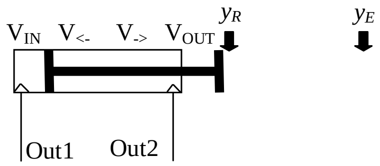


Fig. 4. Plant Element of *Y* axis

The Fig. 5 and Fig. 6 illustrate the determination of all possible abnormal states. For example, when the state q_0 of the PoP sensor y_R model is active, (Fig. 5), the retracted position sensor can be blocked ($B_{/yR}$). For the the DAC model (Fig. 6), starting from the initial state q_0 and in function of the inputs, the DAC model evolves towards states q_1/V_{IN} or q_3/V_{OUT}. At the state q_1/V_{IN}, representing the piston rod in completely retracted position, the cylinder can be blocked in forward direction B_{Vout}. At the state q_3/V_{OUT}, the cylinder can be blocked in backward direction B_{Vin}. All possible abnormal states of the “*Y* axis” PE is classified in Table 2.

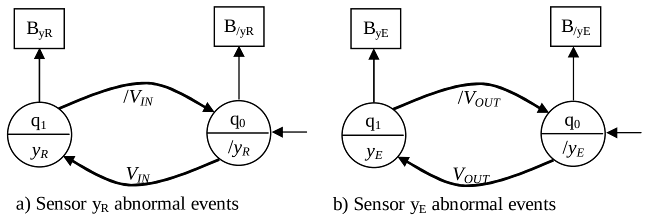


Fig. 5. Determination of sensors y_R and y_E abnormal states

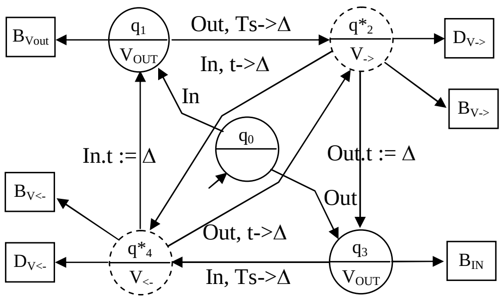


Fig. 6. Determination of DAC abnormal states

Table 2. Classification of abnormal events

Type	Label	Description
Faulty events	B_{yR}	sensor y_R blocked at 1
	$B_{/yR}$	sensor y_R blocked at 0
	B_{yE}	sensor y_E blocked at 1
	$B_{/yE}$	sensor y_E blocked at 0
	B_{Vin}	DAC blocked in retracted direction
	$B_{V->}$	DAC blocked during forward movement
	B_{Vout}	DAC blocked in extended direction
	$B_{V<-}$	DAC blocked during backward movement
Degraded events	$D_{V->}$	DAC too slowly acting in extended direction
	$D_{V<-}$	DAC too slowly acting in retracted direction

3.2 Extrapolation of Gaussian learning

The "*Y* axis" PE can be represented as a block for which the inputs are the control signals of the controller, *In* and *Out*, and the outputs are the sensors' information, y_R and y_E (Fig. 7). The controller is supposed to be safety and dependable. Consequently, it is not possible to have the activation of *In* and *Out* at the same time. When the control signal *Out* is activated, the normal response is $\downarrow y_R$ followed by $\uparrow y_E$. A learning of all sensors' outputs' events can be achieved. For example, Fig. 8 presents the learning of $\uparrow y_E$ after the activation of *Out*. Fig. 9 presents the extrapolation of Gaussian learning when the command *Out* is activated. This activation expects as normal response the sensors events $\downarrow y_R$ and $\uparrow y_E$ within, respectively, the time intervals δ_1 and δ_2. Any other response to this control signal activation will be considered as abnormal behavior. A tolerance interval δ_3 is added to δ_1 and δ_2, in order to take into account the delay in events' occurrences. This interval is subjective and represents about 25 per cent of δ_2. In the next paragraph, the fault candidates' tree is constructed for the "*Y* axis" PE after the activation of a control signal.

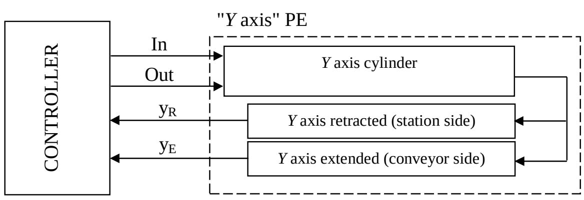


Fig. 7. Observable events of the "*Y* axis" PE

3.3 Candidates for the diagnosis of the "*Y* axis" PE

To define the fault candidates for a PE after the activation of each control signal, all the events consequences are analyzed based on the PoP extended models and their links. For example, Fig. 10 presents the fault candidates' tree after the activation of Out for the "Y axis" PE. If the time delay occurrence of $\downarrow y_R$ belongs to δ_1 and the one of $\uparrow y_E$ to δ_2, then the behavior is considered as normal and the returned label is N (state q_{10}). While if this delay time occurrence of $\downarrow y_R$ and the one of $\uparrow y_E$ belong, respectively, to δ_1 and δ_3 (state q_{11}), then the behavior is degraded and the candidate is: too slowly acting DAC in comparison with its normal displacement velocity (DV->). While if the delay time occurrence of $\downarrow y_R$ belongs to δ_1 and the one of $/y_E$ is greater than δ_3, then the fault candidates are {B/y_E, BV->} (state q12). However, Gaussian learning indicates that B/y_E is more probable than BV->. These fault candidates are proposed to the user in order to facilitate his task of making decision about the system behavior status. The generation of the other candidates' in response to the other control signals activation is achieved similarly.

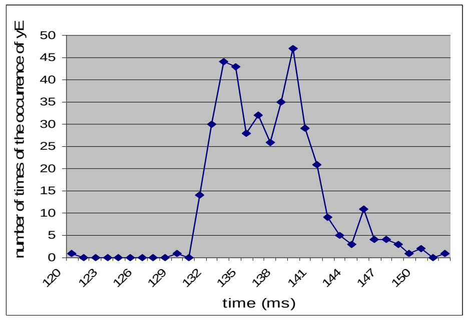


Fig. 8. Learning of the interval of occurrence of the event $\uparrow y_E$ as well as its probability after the activation of Out

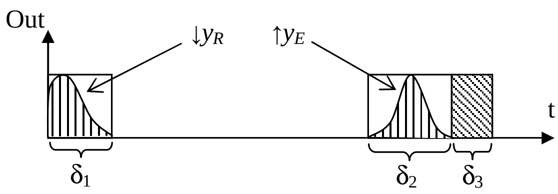


Fig. 9. Extrapolation for the sensors events occurrences intervals after the activation of the control signal Out

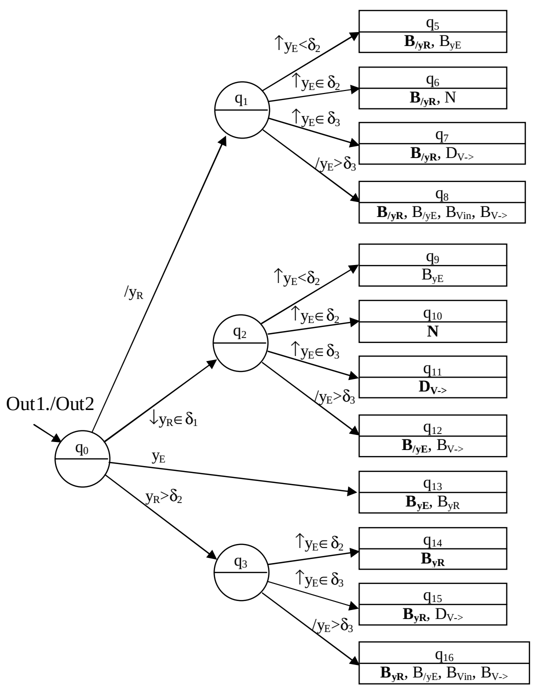


Fig. 10. Fault candidates' tree for the "*Y* axis" PE after the activation of Out

3.4 Remarks

Candidates' tree is only information on the system behaviour. It corresponds to a proposition to the user which is the only one who can take a decision. Consequently, it is used on-line when a command is sent and reinitialized from another command if the behaviour is normal or by the user for an abnormal behaviour.

4. CONCLUSIONS AND FUTURE WORKS

This paper presents a model-based Fault Detection and Isolation (FDI) approach for the diagnosis of discrete manufacturing system. The global model of the system is described by a set of independent plant elements. Each one of the later is composed of a set of interrelated parts of plant. A part of plant can be an actuator or a sensor. The diagnosis is thus distributed on each element plant.

However, only the faults related to PoPs (actuators and sensors) are considered. To take into account the product faults, a product model is necessary. This model depends on the product nature and on the production objective. Thus, a future work is to extend this approach to include the faults related to products. Another perspective is to extend the PoP library by integrating a new family of PoPs with their pre-actuators. The goal is to obtain more realistic plant model.

ACKNOWLEDGEMENTS

This work is integrated in the project MOSYP (Performances Measurements and Optimization of Production Systems). The authors would like to thank the region Champagne-Ardennes within the project MOSYP (CPER ICOS).

REFERENCES

Balemi, S., Hoffmann, G.J., Gyugyi, P., Wong-Toi, H. and Franklin G.F. (1993). Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, vol. 38, N°7, pp.1040-1059.

Boel, B.K. and Jiroveanu G. (2004). Distributed contextual diagnosis for very marge systems. *7th International Workshop on Discrete Event Systems WODES'04*, pp.343-348, Reims, France.

Cassandra, C.G. and Lafortune, S. (1999). Introduction to Discrete Event Systems. *Kluwer Academic Publisher*, ISBN 0792386094.

Chandra, V. and Kumar, R. (2001). A Discrete Event Systems Modeling Formalism Based one Event Occurrences Rules and Precedences. *IEEE Transactions on Robotics and Automation*, vol. 17, n°6.

Cordier, M.O. and Dousson, C. (2000). Alarm driven monitoring based on chronicles. *Proceedings of Safeprocess'2000*, pp. 286–291, Budapest, Hungary.

Cordier, M.O. and Grastien, A. (2007). Exploiting Independence in a Decentralised and Incremental. *20th International Joint Conference on Artificial Intelligence*, pp. 292-297.

Darkhovski, B. and Staroswiecki, M. (2003). Theoretic Approach to Decision in FDI. *IEEE Transactions on Automatic Control*, vol. 48, n°5.

Deluche, J.B. (1998). Automatique de la théorie aux applications industrielles. Tome 1 : Systèmes discrets. *Edipol – Collection Instrumentation*, ISBN 2913444008.

Genc, S. and Lafortune, S. (2003). Distributed diagnosis of discrete-event systems using Petri nets. *International Conference on Application and Theory of Petri Nets*, pp. 316–336, Eindhoven, The Netherlands.

Hadjicostis, C.N. (2005). Probabilistic Fault Detection in Finite-State Machines Based on State Occupancy Measurements. *IEEE Transactions on Automatic Control*, vol. 50, N°12, pp. 2078-2083.

Holloway, L.E. and Chand, S. (1994). Time templates for discrete event fault monitoring in manufacturing systems. *American Control Conference*, Baltimore, USA.

Lamperti, G., Zanella, M. (2008). On Processing Temporal Observations in Monitoring of Discrete-Event Systems. *Book Chapter, Enterprise Information Systems: Part 3*, vol. 3, pp.135-146, ISBN 978-3-540-77580-5.

Machado, J. (2006). Influence de la prise en compte d'un modèle de processus en vérification formelle des systèmes à événements discrets. *Thèse de l'Ecole Normale Supérieure de Cachan*, Cachan, France.

Duda, R. O., Hart, P. E.,. Stork D. G, (2001). Pattern Classification 2nd edition. *Wiley-Interscience*.

Philippot, A., Sayed-Mouchaweh, M. and Carré-Ménétrier, V. (2007). Unconditional Decentralized Structure for the fault diagnosis of Discrete Event Systems. *1st IFAC Workshop on Dependable Control of Discrete-event Systems (DCDS'07)*, Cachan, France.

Philippot, A., Sayed-Mouchaweh, M. and Carré-Ménétrier, V. (2008). Nouvelle modélisation de Partie Opérative pour systèmes manufacturiers. *Conférence Internationale Francophone d'Automatique (CIFA'08)*, Bucarest, Romania.

Qiu, W. (2005). Decentralized/distributed failure diagnosis and supervisory control of discrete event systems. *PhD of the Iowa State University*, USA.

Rohée, B., Riera, B., Carré-Ménétrier, V. and Roussel, J.M. (2006). A methodology to design and Check Plant Model. *3rd IFAC Workshop on Discrete-Event System Design (DESDes06)*, Rydzyna, Pologne.

Rozé, L. and Cordier, M.O. (2002). Diagnosing Discrete-Event Systems: Extending the "Diagnoser Approach" to Deal with Telecommunication Networks. In Discrete Event Dynamic Systems, vol. 12, n°1, pp.43-81, ISSN 0924-6703.

Sampath, M. (1995). A Discrete Event Systems Approach to Failure Diagnosis. *Thesis, University of Michigan*, Michigan, USA.

Wang, Y., Yoo, T.S. and Lafortune S. (2005). Decentralized diagnosis of discrete event systems using conditional and unconditional decisions. *Proceeding of CDC'05, 44th IEEE Conference on Decision and Control*.

Wonham, W. M. and Ramadge, P.J. (1987). On the supremal controllable sublanguage of a given language. *SIAM Journal on Control and Optimization*, vol. 25, n°3, pp.637-659.