



HAL
open science

Time-Based Estimator for Control Reconfiguration of Discrete Event Systems (DES)

Imane Tahiri, Philippot Alexandre, Véronique Carré-Ménétrier, Abdelouahed
Tajer

► **To cite this version:**

Imane Tahiri, Philippot Alexandre, Véronique Carré-Ménétrier, Abdelouahed Tajer. Time-Based Estimator for Control Reconfiguration of Discrete Event Systems (DES). International Conference on Control, Decision and Information Technologies (CoDIT), 2019, Paris, France. 10.1109/CoDIT.2019.8820585 . hal-02114270

HAL Id: hal-02114270

<https://hal.science/hal-02114270>

Submitted on 29 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Time-Based Estimator for Control Reconfiguration of Discrete Event Systems (DES)

I. TAHIRI, A. PHILIPPOT, V. CARRE-MENETRIER, and A. TAJER

Abstract— In this paper, we propose a Time-based estimator method to reconfigure manufacturing systems in case of sensor faults detection. The main idea is to replace the information lost by a timed one. In non-faulty behavior, each sensor is defined through two events: activation and deactivation. Our contribution is to define an estimator model of each event based on different clocks to maintain the same desired behavior of the system in a faulty mode (reconfigured).

Keywords: Discrete event systems, control reconfiguration, manufacturing systems, timed discrete event systems, estimation, sensor faults.

I. INTRODUCTION

During the design and development phases of Modern Complex Systems (MCS), a number of checks and tests are undertaken in order to guarantee their operational safety. However, despite all the measures that can be taken to carry out these tests, there is always the possibility of unpredictable behaviors. This is due to their increasing complexity, operating environment or simply to the aging phenomenon of the system's components.

Because of the growing complexity of MCS and integrated technological elements, unpredictable dysfunctions can have unacceptable consequences. One of the major concerns of recent years research is the reconfiguration of MCS especially Manufacturing Systems (MS). In fact, it is no longer a matter of diagnosing fault events that have already occurred, but a matter of predicting them even before they occur. This, in order to take the necessary measures (reconfiguration, fault-tolerance, maintenance ...) to ensure the safe operation of the system.

In this paper, we propose a so-called *time-based estimator* model for reconfiguring the MS when a sensor fault is detected. First of all, an overview of Discrete Event Systems

(DES) is given in the second section. In section 3 we define the reconfiguration process. Section 4 presents our contributions, and the results application is given in section 5. Finally, in section 6 a conclusion and indications of our future works are reported.

II. OVERVIEW

The proposed approach in this manuscript is applied to MS. These systems are considered as a class of Discrete Event Systems (DES). Unlike continuous systems which are described by differential equations, a DES is defined as a dynamic system that can be described in discrete state space and whose evolution is described by state transitions triggered by events [1]. Two formal methods are proposed in literature to control DES: The first one is the Validation and Verification control (V&V) which aims to let the control designer develop control models based on specifications and then automatically analyses a formal representation of these models [2] and to check if the intended purpose is achieved [3]. The second one is the supervisory control of DES initiated by Ramadge and Wonham [4]. The Supervisory Control Theory (SCT) (figure 1) consists on the one hand in constructing a plant model to represent the possible behaviors of the studied system and on the other hand the specifications model, representing the expected properties. From these two models, the next step is to mathematically obtain the plant model under control whose behavior represents what is possible in the process while respecting the constraints imposed by the specifications. Finally, the supervisor can be synthesized. One of the SCT founding concepts is the notion of *controllability*. Associated with the event concept, this property allows the supervisor to prohibit or not an event, and thus a possible behavior of the process. The major disadvantages of V&V method are the need of prior writing of programs and the impossibility of a programmer to fully

I. TAHIRI is with the Research Center in Information and Communication Science and Technology (CReSTIC), University of Reims Champagne-Ardenne (URCA), France. And she is with the Laboratory of Electrical Engineering and Control of Systems LGECoS, University Cadi Ayyad - ENSA Marrakech, Morocco. (e-mail: imane.tahiri@univ-reims.fr).

A. PHILIPPOT is with the Research Center in Information and Communication Science and Technology (CReSTIC), University of Reims Champagne-Ardenne (URCA), France. (e-mail: alexander.philippot@univ-reims.fr).

V. CARRE MENETRIER is with the Research Center in Information and Communication Science and Technology (CReSTIC), University of Reims Champagne-Ardenne (URCA), France. (e-mail: veronique.carre@univ-reims.fr).

A. TAJER is with the Laboratory of Electrical Engineering and Control of Systems LGECoS, University Cadi Ayyad - ENSA Marrakech, Morocco. (e-mail: a.tajer@uca.ma).

validate it when a change is carried out. Thence, our work is based on the SCT method.

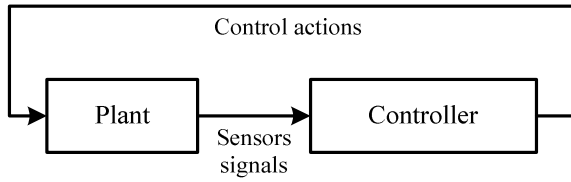


Figure 1: Supervisory Control Theory loop

The original language of the SCT is the formal language generated by Finite State Machines [5]. Formally, an FSM is given as follows:

An FSM is an automaton defined as $A = (Q, \Sigma, \delta, q_0, Q_m)$ such as:

- Q is a finite set of all states of A .
- Σ is the set of events, also called alphabet of A .
- δ is the transition function, defined by $\delta: Q \times \Sigma \rightarrow Q$. $\delta(q, \sigma) = q'$ represents a transition carrying the event σ which leads from a state q to a state q' .
- q_0 is the initial state of the automaton A , such that $q_0 \in Q$.
- Q_m is the set of marked states in A , such that $Q_m \subseteq Q$.

One of the most applied functions on FSM is the synchronous composition which allows representing in a single automaton the parallelized behavior of the automata used for its construction. Let A_1, A_2 denote two automata presenting two DES such as $A_1 = (Q^1, \Sigma^1, \delta^1, q_0^1, Q_m^1)$ and $A_2 = (Q^2, \Sigma^2, \delta^2, q_0^2, Q_m^2)$ with $\Sigma^1 \cap \Sigma^2 \neq \emptyset$. The synchronous composition is given by: $A_3 = A_1 \parallel A_2 = (Q^1 \times Q^2, \Sigma^1 \cap \Sigma^2, \delta^1 \times \delta^2, q_0^1 \times q_0^2, Q_m^1 \times Q_m^2)$.

III. RECONFIGURATION PROCESS

The concept of Reconfigurable Manufacturing Systems (RMS), that was proposed by the University of Michigan in 1999, was the effective solution adopted by researchers to meet the market's requirements and to gain competitiveness. It is necessary to design easily reconfigurable MS in order to adapt them quickly to a changing request (fault detection, configurations changing, or specifications changing) [6], [7] [8], [9], [10] and [11]. This, therefore, requires having systems that are both reconfigurable at the hardware level but also at the control level as solutions to respond to these requirements.

The reconfiguration process [12] is divided into two distinct axes (figure 2): a plant reconfiguration (*hardware*), or a control reconfiguration (*software*). The first axis consists in intervening on the material plane of the plant in order to modify its capacities [13], [14]. It is then possible to compensate the services lost following the occurrence of a failure by the introduction following the reconfiguration of new services. While the second solution is to have a reconfigurable control able to adapt and exploit the services still available offered by the operative part. Several approaches have been proposed in [15] [16], [17], [18], [19] and [20].

In this work, we focus on the second solution to maintain the MS functioning despite faults that can hinder its normal

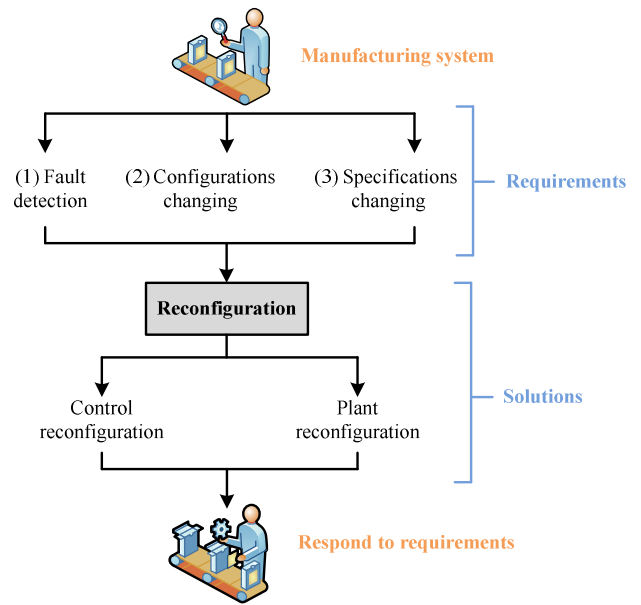


Figure 2: Reconfiguration process

behavior. In order to obtain a reconfigurable control, two solutions are proposed in the literature:

- Online synthesis of a deterministic control law exploiting the capacity of the operative part [21] and [22].
- Online adaptation of the control developed offline, based on the idea that all the potentialities of the operative part have been pre-integrated to the control [23].

Whether it is a plant or a control reconfiguration, the process of reconfiguration is necessarily based on a knowledge of the services lost following the occurrence of a symptom of failure. This knowledge is usually derived from the diagnostic function (figure 3) which aims to locate and then identify the cause of the failure. It should be noted here that the role of such a diagnostic function in the reconfiguration process is not only to identify the cause of the failure for a maintenance purpose but also to identify the failure and its consequences directly or indirectly on the capacity of the plant and on the product flow state, with the aim of updating the operational capabilities model of the plant.

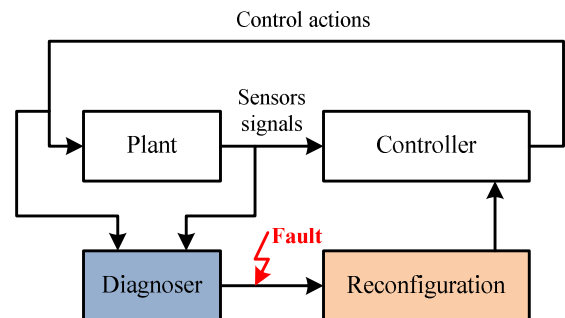


Figure 3: Control reconfiguration's loop

IV. PROPOSED APPROACH

In this paper, a new framework of control reconfiguration is proposed. The method is based on the definition of a *time-based estimator* model (faulty behavior). The main idea is to reconfigure the normal behavior model (Controller_(NB)) by a faulty one (Controller_(FB)) in the case of sensor faults detection (figure 4). Indeed, the information lost about the faulty sensor is replaced by timed information to maintain the system's functioning. We note that the diagnosis task is not discussed in this paper.

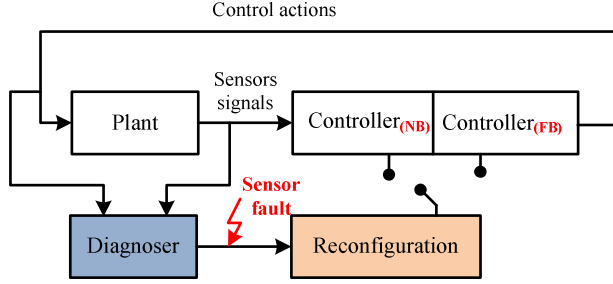


Figure 4: Control reconfiguration loop with two controllers

Several approaches discussed the control reconfiguration issue for a centralized [24], hierarchical [25], decentralized [26], or distributed control [27], [28]. Although these approaches may have solved the reconfiguration problem, they nevertheless rely on material redundancy to fill the failure of a system component. This strategy is not only expensive, but it also requires a major maintenance device. Moreover, they remain dependent on Boolean information.

But it is not the only information present when we consider time. This is called Timed Discrete Event Systems (TDES). In literature, time is modeled in two ways. The first one is the discrete time, for which the progression of time is discrete and measured by a digital clock which is incremented by one unit at a time, and the second one is the dense time, for which the progression of time is continuous and measured by clocks taking positive real values. TDES can be modelled using timed automata, Petri nets, or finite state machines [29], [30],[31], [32] and [33]. In a previous work, we discussed a method of timed synthesis control of TDES. A method where time is presented through a clock and considered as an event, which makes the modeling phase by FSM a simple task. In this work, a TDES is described by the following automaton: $A = (Q, \Sigma, \delta, q_0, Q_m)$ such as:

- Q is a finite set of all states of A .
- Σ is the set of events, such as $\Sigma = \Sigma_{nT} \cup \Sigma_T$. With: Σ_{nT} is the set of non-timed events and Σ_T is the set of timed events such as: $\Sigma_T = C \cup D$ with:
 - C : Set of clocks, each clock is defined by an activation and deactivation $C = \uparrow c_i \cup \downarrow c_i$
 - D : Finite set of durations d_i associated to each clock c_i , such as $D = \{d_1, d_2, \dots, d_i\}$.
- δ is the transition function. A transition is defined by: $\delta(q, \sigma) = q'$. σ is the occurrence of a timed event or not of Σ .
- q_0 is the initial state of the automaton A , such that $q_0 \in Q$.
- Q_m is the set of marked states in A , such that $Q_m \subseteq Q$.

In this paper, we do not go into the details of the control synthesis. We focus on how to define a model to replace the information lost on the sensors during the fault's detection. We note that the faults treated in the rest of the paper are non-observable faults of sensors. A non-observable fault is detected when a sensor is stuck in a position 0 or 1.

A boolean sensor can take two values (figure 5): (=1) if the sensor is activated or (=0) if it is not. Both values are reached through the events associated to the sensor. Let "s" denote the boolean sensor, its associated events are " $\uparrow s$ " (activation) and " $\downarrow s$ " (deactivation).

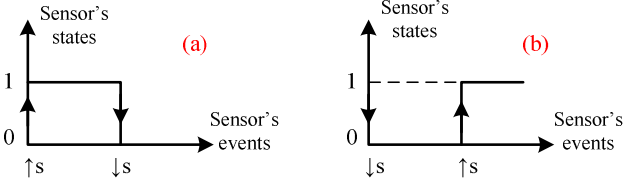


Figure 5: Sensor's states and events

To model the sensor functioning of an MS, we opt for the practical modeling proposed in [34]. The author separates the MS into several Plant Elements (PE). Then he considers two models: the first one noted (actuator model) describes the functioning of each actuator with respect to its associated detectors. The second one noted (detectors) describes only the functioning of detectors associated to each PE. Finally, the synchronization of these two models allows to obtain an automaton that describes all possible behaviors of the actuator with its detectors.

The two functioning modes of sensor "s" shown in Figure 5 can be modeled by the following FSMs:

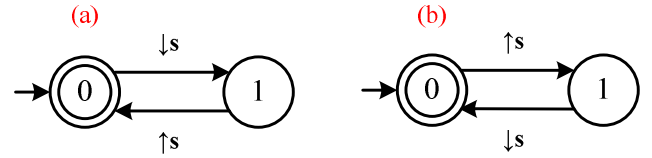


Figure 6: Sensor models

The FSM_(a) (figure 6) describes the sensor "s" behavior when the initial state of "s" is activated and the FSM_(b) presents the sensor "s" behavior when its initial state is deactivated.

If the diagnoser detects a fault on the sensor "s", the normal behavior is disrupted, and the models shown in figure 6 are not working anymore. Hence, the replacement of the sensor information is necessary. Therefrom, we propose a model estimating the functioning of the sensor and which ensures the same desired behavior (non-faulty behavior) of the system. The proposed model is a *time-based estimator* where the sensor activation " $\uparrow s$ " is replaced by a clock " C_i ", and the deactivation " $\downarrow s$ " by a clock " C_j ". Each clock is defined through three events: the clock activation $\{\uparrow c_i, \uparrow c_j\}$, the clock deactivation $\{\downarrow c_i, \downarrow c_j\}$, d_i the necessary duration for "s" to turn on and d_j the necessary duration of "s" to turn off. The durations are estimated and determined by learning.

The corresponding estimator model of figure 6 is given then as follows (figure 7):

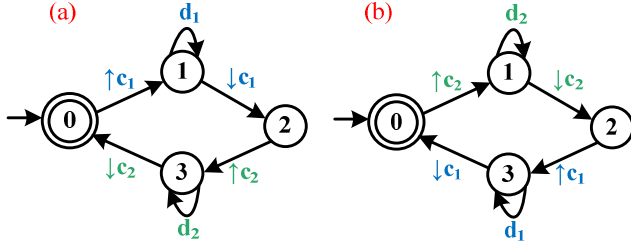


Figure 7: Time-based estimators

If c_1 or c_2 is enabled, the system remains blocked in the state (1) for a duration d_1 or d_2 which is expressed by a self-loop transition on state (1). Once it is elapsed, then the clock deactivation is allowed.

The purpose behind the design of *time-based estimator* models is the control reconfiguration of normal behavior when a fault is detected. The control is based on SCT to determine the supervisors which comply with specifications. Indeed, the models that we determine describe all the possible behaviors. From these models, the SCT allows the determination of desired behavior and prohibits the undesirable situations. The resulting models of SCT allow the maintain of the system functioning despite the fault detected

V. APPLICATION

In this section, we present a case study to reveal and evince the effectiveness of our contribution. To this aim, we apply the proposed approach to the monostable single effect cylinder shown in figure 8.

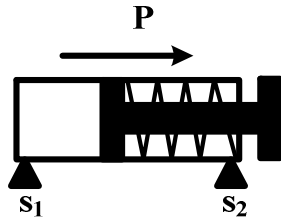


Figure 8: Monostable single effect cylinder

Two limit sensors are associated to the cylinder s_1 and s_2 . The exit order is given by the action P . Three cases are studied:

- No sensor faults are detected (normal behavior)
- One sensor fault is detected
- Two sensor faults are detected

The actuator functioning with its detectors is as follows: In the rest state, the spring is released and the sensor “ s_1 ” is activated. Once the order of exit “ P ” of the stem is authorized, “ s_1 ” is deactivated and the order is maintained until the activation of sensor “ s_2 ” which causes the deactivation of the order and then the stem’s return to its initial state by the recall of the spring.

A. Normal behavior

In normal behavior (figure 9), both sensors s_1 and s_2 are working. Applying the practical model proposed for non-faulty behavior in [34], we obtain the two following models (1) and (2):

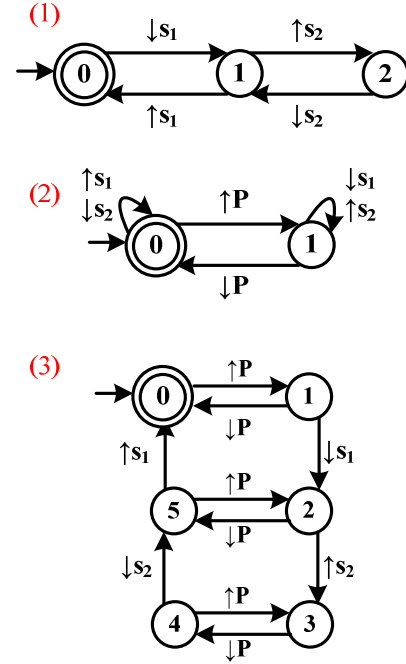


Figure 9: (1) detectors model, (2) actuator model, (3) Plant Element model

By synchronizing the two models (1) and (2), we obtain the model (3) that describes all possible evolutions of the cylinder with its associated sensors.

The model (3) automaton is defined as follows:

$A_{(3)} = (Q, \Sigma, \delta, q_0, Q_m)$ such as:

- $Q = \{0, 1, 2, 3, 4, 5\}$
- $\Sigma = \{\uparrow P, \downarrow P, \uparrow s_1, \downarrow s_1, \uparrow s_2, \downarrow s_2\}$
- $q_0 = 0$.
- $Q_m = 0$.

B. Faulty behavior (one fault)

We assume that s_2 is faulty. Let C_1 and C_2 replace respectively the sensor s_2 activation and deactivation. The corresponding estimator models are given by figure 10.

The model (3) automaton is defined as follows:

$A_{(3)} = (Q, \Sigma, \delta, q_0, Q_m)$ such as:

- $Q = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{\uparrow P, \downarrow P, \uparrow s_1, \downarrow s_1, \uparrow c_1, \downarrow c_1, d_1, \uparrow c_2, \downarrow c_2, d_2\}$
- $q_0 = 0$.
- $Q_m = 0$.

In model (2) automaton, the events associated to sensor s_2 ($\uparrow s_2$ and $\downarrow s_2$ resp) are replaced by their estimated events ($\{\uparrow c_1, \downarrow c_1, d_1\}$ and $\{\uparrow c_2, \downarrow c_2, d_2\}$ resp) on the self-loop transition labels.

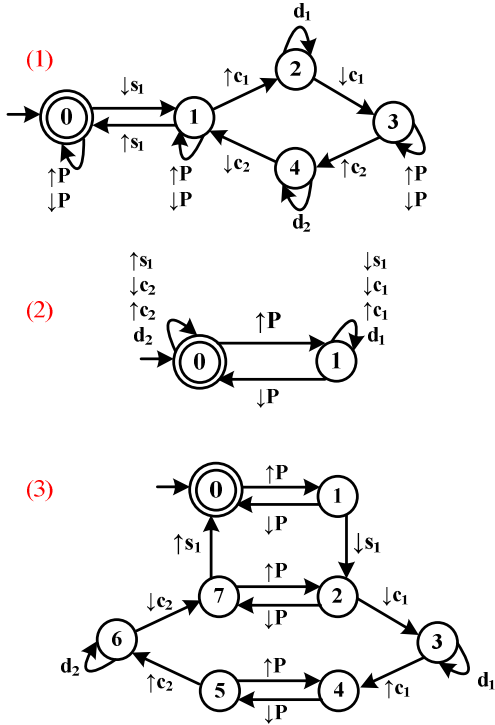


Figure 10: Estimators models (1) detectors model, (2) actuator model, (3) Plant Element model

In the initial state, the cylinder is in the input position and the sensor s_1 is activated. Once the output order $\uparrow P$ is allowed, the sensor s_1 is deactivated “ $\downarrow s_1$ ” and the clock C_1 is triggered for a duration d_1 which is the estimated time for the cylinder to reach the complete exit position (corresponding to s_2 activation in normal behavior). If d_1 is not elapsed, the exit order is maintained until the flow of d_1 . Once the C_1 is deactivated, the exit order is released and then the clock C_2 is activated for a duration d_2 (corresponding to s_2 deactivation in normal behavior).

C. Faulty behavior (two faults)

We assume that s_1 and s_2 are faulty. Let C_3 and C_4 replace respectively the sensor s_1 activation and deactivation. Let C_1 and C_2 replace respectively the sensor s_2 activation and deactivation. The corresponding estimator models are given by figure 11.

In the initial state, if the output order $\uparrow P$ is allowed, the clock c_4 is activated for a duration d_4 which is the estimated time for s_1 to be deactivated. Then, the clock C_1 is triggered for a duration d_1 which is the estimated time for the cylinder to reach the complete exit position (corresponding to s_2 activation in normal behavior).

The model (3) automaton is defined as follows:

$A_{(3)} = (Q, \Sigma, \delta, q_0, Q_m)$ such as:

- $Q = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\Sigma = \{\uparrow P, \downarrow P, \uparrow c_1, \downarrow c_1, d_1, \uparrow c_2, \downarrow c_2, d_2, \uparrow c_3, \downarrow c_3, d_3, \uparrow c_4, \downarrow c_4, d_4\}$
- $q_0 = 0.$
- $Q_m = 0.$

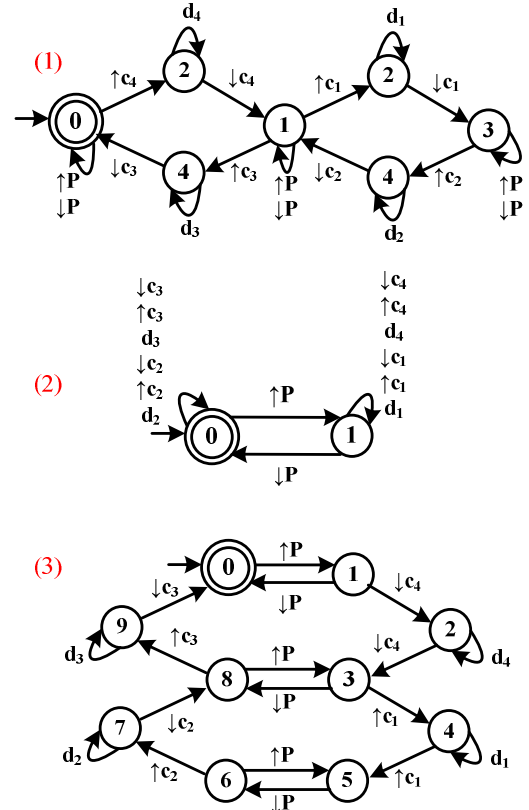


Figure 11: Estimators models (1) detectors model, (2) actuator model, (3) Plant Element model

In model (2) automaton, the events associated to sensors s_1 ($\uparrow s_1$ and $\downarrow s_1$ resp) and s_2 ($\uparrow s_2$ and $\downarrow s_2$ resp) are replaced by their estimated events ($\{\uparrow c_3, \downarrow c_3, d_3\}$ and $\{\uparrow c_4, \downarrow c_4, d_4\}$ resp) and ($\{\uparrow c_1, \downarrow c_1, d_1\}$ and $\{\uparrow c_2, \downarrow c_2, d_2\}$ resp) on the self-loop transition labels.

Once C_1 is deactivated, the exit order is released and then the clock C_2 is activated for a duration d_2 (corresponding to S_2 deactivation in normal behavior). After C_2 deactivation, C_3 is triggered for a duration d_3 which is the estimated time for the cylinder to reach its initial position (corresponding to s_1 activation in normal behavior).

VI. CONCLUSION AND PROSPECTS

In this manuscript, we have proposed a new approach of integrating time in order to define a time-based estimator for discrete event systems DES. This estimator aims at achieving a control reconfiguration during the detection of a sensor fault thanks to the diagnoser. Indeed, the information lost on the sensor is replaced by timed information. The main idea is to define an estimated model for sensor activation and deactivation based on clocks. Each clock is associated to three events: its activation, deactivation and its duration which is the estimated time for a sensor “ s ” to be activated or not. The results are applied to a monostable simple effect cylinder in order to prove the relevance of our method.

The key advantage of our approach is the use of separate plant element models in order to exploit a distributed control that in the one hand avoid the combinatorial explosion

recurrent in the reconfiguration approaches based on centralized and decentralized control [20], [19], [18]. On the other hand, it allows the reconfiguration of the only faulty plant element without reconfiguring all the system's control. In addition, to replace the faulty sensor events by timed events that ensure the same behavior avoid the use of redundant elements which is an expensive solution. The restriction of the approach lies on the use of boolean sensors that have two states: sensor=1 (activation) or sensor=0 (deactivation), this type of sensors can be replaced by timed properties as we showed before which is not the case if we discuss analogic sensors. Certainly, the parallel execution of the two operating modes (normal and reconfigured) requires additional resources of memory and calculation as well as a robust synchronous communication. But it also minimizes the analysis time of a new fault and offers constantly a new alternative ready to be used when it is needed.

In our short-term works, a library of time-based estimators of different type of cylinders with their associated sensors will be proposed. Furthermore, an approach of control reconfiguration of DES based on SCT will be proposed too, the method is based on a distributed control exploiting the fact that MS are informally distributed [28].

REFERENCES

- [1] C. G. Cassandras et S. Lafortune, *Introduction to Discrete Event Systems*, Second Edition. New York, 2008.
- [2] J.-M. Roussel, J.-M. Faure, J.-J. Lesage, et A. Medina, « Algebraic approach for dependable logic control systems design », *Int. J. Prod. Res.*, vol. 42, n° 14, p. 2859-2876, juill. 2004.
- [3] T. M. Press, « Principles of Model Checking », *The MIT Press*. [En ligne]. Disponible sur: <https://mitpress.mit.edu/books/principles-model-checking>. [Consulté le: 08-janv-2019].
- [4] P. J. G. Ramadge et W. M. Wonham, « The control of discrete event systems », *Proc. IEEE*, vol. 77, n° 1, p. 81-98, janv. 1989.
- [5] M. Sipser, *Introduction to the Theory of Computation, Second Edition*. Course technology, 2005.
- [6] S. T. Mortensen et O. Madsen, « Operational Classification and Method for Reconfiguration & Recommissioning of Changeable Manufacturing Systems on System Level », *Procedia Manuf.*, vol. 28, p. 90-95, janv. 2019.
- [7] Y. Koren *et al.*, « Reconfigurable Manufacturing Systems », *CIRP Ann.*, vol. 48, n° 2, p. 527-540, janv. 1999.
- [8] M. G. Mehrabi, A. G. Ulsoy, et Y. Koren, « Reconfigurable manufacturing systems: Key to future manufacturing », *J. Intell. Manuf.*, vol. 11, n° 4, p. 403-419, août 2000.
- [9] Y. Koren, « General RMS Characteristics. Comparison with Dedicated and Flexible Systems », in *Reconfigurable Manufacturing Systems and Transformable Factories*, A. I. Dashchenko, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, p. 27-45.
- [10] M. Bortolini, F. G. Galizia, et C. Mora, « Reconfigurable manufacturing systems: Literature review and research trend », *J. Manuf. Syst.*, vol. 49, p. 93-106, oct. 2018.
- [11] A. Molina *et al.*, « Next-generation manufacturing systems: key research issues in developing and integrating reconfigurable and intelligent machines », *Int. J. Comput. Integr. Manuf.*, vol. 18, n° 7, p. 525-536, oct. 2005.
- [12] P. BERRUET, J. Petin, F. RIGAUD, A. TOGUYENI, et E. ZAMAI, « Architectures de pilotage de procédés industriels », *Techniques de l'ingénieur. L'Entreprise industrielle*, p. numéro AG 3510, pages 1-19, 2007.
- [13] R. G. Landers, B.-K. Min, et Y. Koren, « Reconfigurable Machine Tools », *CIRP Ann.*, vol. 50, n° 1, p. 269-274, janv. 2001.
- [14] E. Travaini, P. Pedrazzoli, R. Rinaldi, et C. R. Boër, « Methodological Approach and Reconfiguration Tool for Assembly Systems », *CIRP Ann.*, vol. 51, n° 1, p. 9-13, janv. 2002.
- [15] E. Estévez, D. Orive, et M. Marcos, « Model Based Reconfiguration to Controller Failures in Manufacturing Systems », *IFAC Proc. Vol.*, vol. 45, n° 4, p. 31-36, janv. 2012.
- [16] G. Faraut, L. Piétrac, et E. Niel, « Control law synthesis and reconfiguration using SCT », in *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, 2010, p. 576-581.
- [17] R. Kumar et S. Takai, « A Framework for Control-Reconfiguration Following Fault-Detection in Discrete Event Systems* », *IFAC Proc. Vol.*, vol. 45, n° 20, p. 848-853, janv. 2012.
- [18] T. Wittmann, J. H. Richter, et T. Moor, « Fault-Hiding Control Reconfiguration for a Class of Discrete Event Systems », *IFAC Proc. Vol.*, vol. 46, n° 22, p. 49-54, janv. 2013.
- [19] T. Moor, « A discussion of fault-tolerant supervisory control in terms of formal languages », *Annu. Rev. Control*, vol. 41, p. 159-169, janv. 2016.
- [20] M. Macktoobian et W. M. Wonham, « Automatic reconfiguration of untimed discrete-event systems », in *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2017, p. 1-6.
- [21] S. Henry, E. Deschamps, E. Zamaï, et M. Jacomino, « Control Law Synthesis Algorithm for Discrete-Event Systems », in *IFAC Conference on Management and Control of Production and Logistics (MCPL'04)*, Santiago, Chile, 2004.
- [22] F. F. de Lamotte, P. Berruet, et J.-L. Philippe, « A MODEL FOR THE RECONFIGURATION OF MANUFACTURING SYSTEMS », *IFAC Proc. Vol.*, vol. 38, n° 1, p. 122-127, janv. 2005.
- [23] D. Gouyon, J. Petin, et G. Morel, « Control Synthesis for Product-Driven Automation », *IFAC Proc. Vol.*, vol. 37, n° 18, p. 21-26, sept. 2004.
- [24] C. Seatzu, M. Silva, et J. H. van Schuppen, Éd., *Control of Discrete-Event Systems: Automata and Petri Net Perspectives*. London: Springer-Verlag, 2013.
- [25] B. P. Zeigler, A. Muzy, et E. Kofman, « Chapter 4 - Introduction to Discrete Event System Specification (DEVS) », in *Theory of Modeling and Simulation (Third Edition)*, B. P. Zeigler, A. Muzy, et E. Kofman, Éd. Academic Press, 2019, p. 93-126.
- [26] A. Khoumsi et H. Chakib, « Decentralized supervisory control of discrete event systems moving decisions closer to actions », in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2014, vol. 02, p. 280-287.
- [27] W. M. Wonham et P. J. Ramadge, « Modular supervisory control of discrete-event systems », *Math. Control Signals Syst.*, vol. 1, n° 1, p. 13-30, févr. 1988.
- [28] Y. Qamsane, A. Tajer, et A. Philippot, « A synthesis approach to distributed supervisory control design for manufacturing systems with Grafcet implementation », *Int. J. Prod. Res.*, sept. 2016.
- [29] J. S. Ostroff, « Deciding properties of timed transition models », *IEEE Trans. Parallel Distrib. Syst.*, vol. 1, n° 2, p. 170-183, avr. 1990.
- [30] B. A. Brandin et W. M. Wonham, « Supervisory control of timed discrete-event systems », *IEEE Trans. Autom. Control*, vol. 39, n° 2, p. 329-342, févr. 1994.
- [31] A. T. Sava et H. Alla, « A control synthesis approach for time discrete event systems », *Math. Comput. Simul.*, vol. 70, n° 5, p. 250-26. 2006.
- [32] A. Khoumsi et L. Ouédraogo, « New results in modeling and supervisory control of real-time discrete event systems », in *2010 18th Mediterranean Conference on Control Automation (MED)*, 2010, p. 669-676.
- [33] B. F., C. P., et C. J., « An incremental model repair approach to timed discrete event systems », *IFAC-Pap.*, vol. 50, n° 1, p. 13636-13641, juill. 2017.
- [34] A. Philippot, « Contribution au diagnostic décentralisé des systèmes à événements discrets: Application aux systèmes manufacturiers », Reims champagne Ardenne University, Reims, France, 2006.