Fault-Tolerant Control of Discrete Event Systems: Comparison of two approaches on the same case study

Julien Niguez, Saïd Amari, Jean-Marc Faure

Fault-Tolerant Control of Discrete Event Systems: Comparison of Two Approaches on the same Case Study

J. Niguez, S. Amari and J.-M. Faure
Automated Production Research Laboratory
LURPA, ENS Cachan, Univ Paris Sud F-94235 Cachan, France
Mail: {julien.niguez, said.amari, jean-marc.faure}@ens-cachan.fr

Abstract—**In this paper, two approaches of Fault-Tolerant Control for Discrete Event Systems are compared: the Fault-Hiding approach [4] and the control reconfiguration approach [5]. They are applied on a single case study in order to be compared. Then, a discussion on processing capacities, model sizes and limitations of the two methods is proposed.**

I. Introduction

Productivity within a company is a major challenge, with significant economic implications. In dependability, a high availibity of a system ensures a good productivity. Availability depends among other on the ability of the system to adapt to faults before they have a negative impact on production. Fault-Tolerant Control (FTC) is a means of dependability that allows to interact with the system controller, in order to adapt the control to a faulty behavior of the plant. The production strategy can be accommodated before the productivity of the system is reduced.

The basics of Fault-Tolerant Control for continuous systems are presented in [1]. A definition of fault tolerance is proposed: the ability of a controlled system to maintain control objectives, despite the occurrence of a fault. A degradation of control performance may be accepted. Fault-tolerance can be obtained through fault accommodation or through system and/or controller reconfiguration. Hence methods of fault-tolerant control for Discrete Event Systems (DES) can be classified into two categories: methods based on fault-accommodation ([2], [3], [4]) and methods based on reconfiguration ([5], [6]).

This paper proposes a comparison of the fault-hiding approach [4] and of the control reconfiguration approach [5] through an application on a single case study. These methods were selected in order to compare one method using fault-accommodation and on method using reconfiguration. Futhermore, methods differ by the presence or absence of a diagnoser to detect the occurrence of a fault. It have been chosen to compare a method using a diagnoser with another not using one. Finally, the latest approaches were chosen.

II. Presentation of the case study

The system used for the comparison of selected methods is the sorting system depected in figure 1. The system consists of a loading conveyor (mark A in Figure 1), which routes the boxes to the intermediate conveyor (B). The boxes are then conveyed to a turntable (C), on which there are rollers (D). The unloading conveyors (E and G) are not considered in the following.

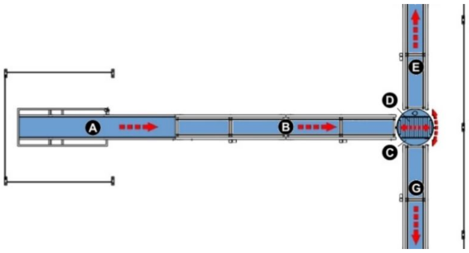

Fig. 1. The sorting system

Boxes of two sizes are delivered to the table: small boxes and big boxes. The objective of the system is to distribute small boxes to the right and big boxes to the left.

A. System operation

To achieve its objective, the system has the following actuators and sensors, respectively presented in Tables I and II. In the following, controllable events of Σ_c (set of controllable events) will be associated to actuators controls and uncontrollable events of Σ_u (set of uncontrollable event) to sensors informations.

TABLE I. Actuators of the system and associated events

Actuator	Event
Conveyor A rotation	A
Conveyor B rotation	B
Rollers rotation - clockwise	R_h
Rollers rotation - counterclockwise	R_a
Table rotation - left	T_g
Table rotation - right	T_d

In models of this paper, rising (respectively falling) edge of a sensor event e will be noted e_1 (e_0) and activation (deactivation) of an actuator E will be noted E_1 (E_0).

The system operates as follows. Boxes arrive randomly on the conveyor A. Once they reach the end of this conveyor, they are sent on conveyor B if free. This conveyor stop when a box arrives at the end of conveyor B but the table is not in the center position. A box is loaded on the table thanks to the clockwise rotation of the rollers. Once the table charged, it is turned to the left position. Large box (small box respectively)

TABLE II. SENSORS OF THE SYSTEM AND ASSOCIATED EVENTS

Sensor	Event
Box at the end of conveyor A	c_a
Box at the end of conveyor B	c_b
Big box loaded on table	c_g
Small box loaded on table	c_p
Table - left position	p_g
Table - middle position	p_m
Table - right position	p_d
Box delivered to the right	d_d
Box delivered to the left	d_g

are then discharged to the left (right) using the clockwise rotation (counterclockwise rotation) of the rollers. The table then returns to the center position, ready to receive a new box.

B. Faults classification

Faults will be considered as non-repairable and modeled by unobservable events. Simultaneous occurrences of faults are not considered.
The following classification of faults is proposed :

- *Actuator faults* are faults involving either an actuator (for instance a motor or a cylinder), a pre-actuator or a connection between an actuator and a pre-actuator.
- *Sensor faults* are faults involving either a sensor or a connection of the input unit of the controller.
- *Process Faults* are faults involving the process in the plant (for instance a box falling down a conveyor)

It is chosen to illustrate the *actuator* faults by a failure of the pre-actuator in charge of the counterclockwise rotation of the conveyor. The *sensor* fault is modeled by a failure of the sensor at the end of conveyor A. However, a *process* fault can not be treated with this kind of modelization. A lack of event is usually handled with the implementation of a timer. Hence, this kind of fault can not be handled through chosen formalisms which are non-timed.

In the following, a fault will be represented by the event f, such that $f \in \Sigma_u$ is non-observable.

C. Reconfiguration strategy

By construction, the systems can achieve its goal even if one of the faults considered above occurs.
In the case of the *actuator* fault, the system operation becomes the following : a box is loaded on the table through the clockwise rotation of the conveyor. Once loaded, the table is rotated to its left position (respectively right) if the charged box is a big box (a small box). Then, the box is distributed by using the clockwise rotation of the table. Finally, the table returns to its middle position.
In the case of the *sensor* fault, it is no longer possible to detect a package at the end of the conveyor A. The system having no redundancy for this function, a degraded behavior is specified in order to avoid collisions between boxes. Conveyor A is stopped as soon as a box is detected at the end of conveyor B and the table is not ready, regardless that there is a package at the end of the conveyor A or not.

III. FAULT-HIDING APPROACH

The Fault-Hiding method was introduced in [4]. A reconfiguration block is interposed between the plant and the controller (see Figure 2). This block aim to interpret exchanges between the plant and the controller. If a fault occurs, the reconfiguration block is allowed to act on the transmitted informations, in order to simulate a controller that can adapt to the fault.

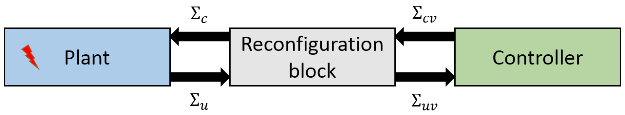

Fig. 2. Closed-loop system in *Fault-Hiding method*

An *virtual* alphabet $\Sigma_v = \Sigma_{cv} \cup \Sigma_{uv}$ is introduced. Depending on the Inputs received (Σ_u and Σ_{cv}), the reconfiguration block adapts the Outputs emitted (Σ_c znd Σ_{uv}).

The originality of this method comes from the following characteristic: it does not require the use of a diagnoser to detect the occurrence of the fault, and does not alter the original models of the plant and the controller.

In this paper, only the construction of the reconfiguration block will be exposed. The model of the controller is assumed to be known. The formalism used to this approach is that of finite state automata.

More details on the formalism and the construction process can be found in [4].

IV. CONTROL RECONFIGURATION APPROACH

The method presented in [5] proposes a technic for control reconfiguration. It principle is shown in Figure 3. Faults are detected, isolated and reported to reconfigurator using a diagnoser. The reconfigurator will then adapt the control law.

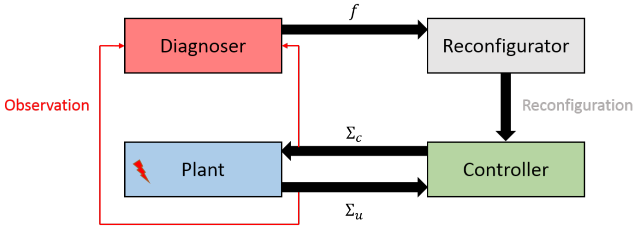

Fig. 3. Closed-loop system in *Control reconfiguration method*

The method provides a reconfiguration strategy of the controller upon detection of a fault on the system. Two methods are proposed:

- Trajectory re-planning: a new trajectory that ensures a correct behavior of the system is selected from the model of the reconfigurator.
- Input/Output adaptation: the input or output event associated with the failed component is substituted by an event that ensures a correct behavior of the system.

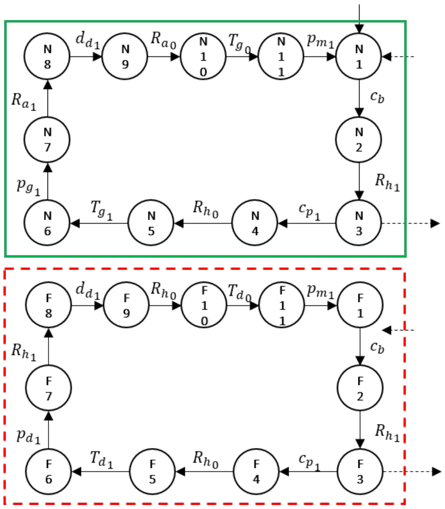

Fig. 4. Model of the accommodation specification for the actuator fault

More details on the formalism and the construction process can be found in [5].

V. Processing of different cases of faults

For space reasons, this section presents only some of the models obtained with the two methods. These models were selected to highlight limitations and special features of both methods.

A. *First case of fault - actuator fault*

The first fault considered is the fault corresponding to the failure of the actuator controlling the table rollers in the counterclockwise direction.

Figure 4 presents a part of the model of the fault-accommodation specification of components $\{roller + table\}$. For the sake of simplicity, only a portion of the figure is presented here: this part corresponds to the management of a small box. The treatment of a large box is similar.

This model built by expert knowledge consists of 2 parts: the part in the green frame is the nominal behavior, while the section in the red frame is the desired behavior after occurrence of the fault. This corresponds to control and reconfiguration strategies presented in part II. Figure 4 shows that there is no transition from the model of the nominal behavior to the model of the faulty behavior. This means that it is not possible to detect the occurrence of the fault by a sequence of events corresponding to a faulty behavior. The system remains blocked, waiting for the event d_d_1) to occur.

Figure 5 presents the model of the reconfigurator of the control reconfiguration method obtained for the processing of the actuator fault. This model is obtained from the previously constructed treillis. For the same reasons than the figure 4, only a portion of the figure 5 is presented.

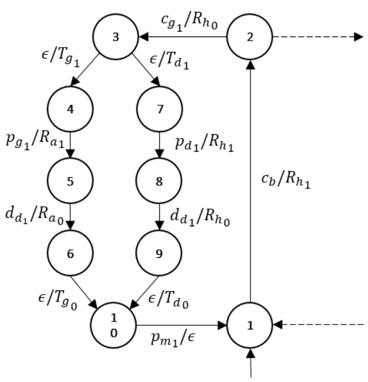

Fig. 5. Model of reconfigurator for the actuator fault

It is possible to extract from the model of the reconfigurator in Figure 5 a control law respectful of the nominal behavior, which here corresponds to the model in Figure 5 without states $\{7, 8, 9\}$. After occurrence of the fault, the latter is reported to the reconfigurator. Then, the trajectory re-planning technic grant a new control law corresponding to the model shown in Figure 5 without states $\{4, 5, 6\}$.

TABLE III. Size of the models for for the processing of the actuator fault (FH: Fault-Hiding, CR: Control Reconfiguration

Models	States	Trans.
FH - Fault-accommodation specification	38	40
FH - Reconfiguration block	-	-
CR - Reconfigurator	18	21

B. *Second case of fault - sensor fault*

The second fault considered corresponds to a failure of the sensor at the end of the conveyor A.

Figure 6 presents a part of the model of the fault-accommodation specification of component $\{conveyorA + conveyorB\}$.

As in the first case, the model in Figure 6 was obtained by expert knowledge, and frames have the same meaning. We can notice in Figure 6 a transition from the model of the nominal behavior to the model of the faulty behavior (labeled by the event f). Even without diagnoser, it is possible to detect the occurrence of fault by a faulty sequence of events, which here corresponds to the occurrence of an event c_b_1 from state 0.

Figure 7 presents the model of reconfigurator obtained for the processing of sensor fault. This model is obtained from the previously constructed trellis.

The model shown in Figure 7 corresponds to the control law respecting the control specification. After occurrence of the fault, the later is reported to the reconfigurator. Then, it is possible to determine a new control law by using the input/output adaptation. In the model in Figure 7, transitions in which appear the event c_a must be adapted as follow: event c_a_0 is changed into the event 0 (transition is never validated) and c_a_1 into 1 (transition is always validated). For example, the transition $\delta(1, c_{a\ 0}, .) = 2$ becomes $\delta(1, 0, .) = 2$.

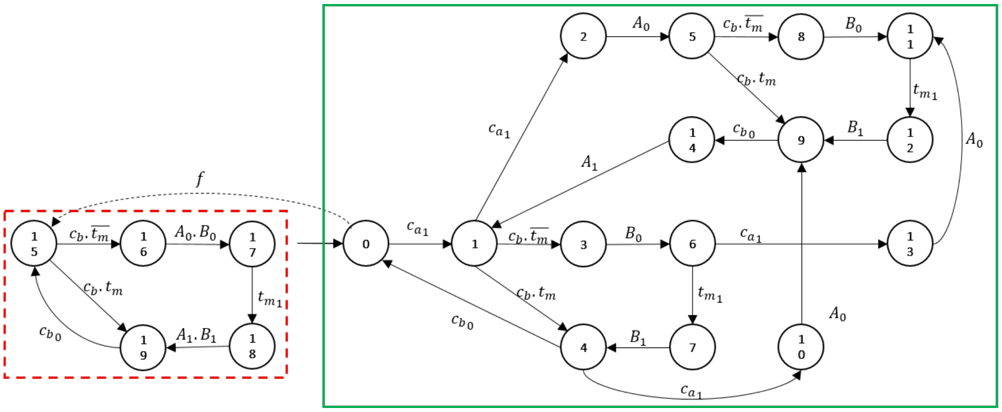

Fig. 6. Model of the accommodation specification for the sensor fault

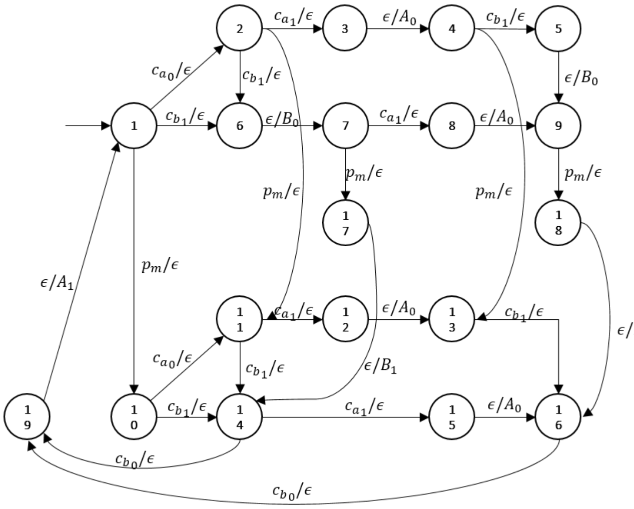

Fig. 7. Model of reconfigurator for the sensor fault

TABLE IV. SIZE OF THE MODELS FOR FOR THE PROCESSING OF THE SENSOR FAULT (FH: FAULT-HIDING, CR: CONTROL RECONFIGURATION

Models	States	Trans.
FH - Fault-accommodation specification	19	27
FH - Reconfiguration block	52	75
CR - Reconfigurator	19	29

VI. DISCUSSION AND CONCLUSION

About the processing capacity of the fault-hiding approach, it was impossible to obtain a model of the reconfiguration block for the actuator fault. Indeed, it is not possible to detect the occurrence of the fault, and thus to adapt the control. For the second type of fault however, a faulty sequence of events allow the detection of the fault, hence the reconfiguration block is abled to achieve fault-tolerant control. It can also be noted that the construction of the reconfiguration block requires to model faults and the behavior of the system after occurrence of faults, which expose the method to a problem of space explosion in case of scaling.

Concerning the reconfiguration method, it was possible to obtain a fault-tolerant control in both fault cases. From the model of the reconfigurator, trajectory replanning has resulted in a new control law using the redundancy of the system in the first case. For the second type of errors, input/output adaptation allowed to adapt the control law so as to reach a performance-degraded behavior after occurrence of the fault. However, the trellis necessary to the construction of the reconfigurator were made by hand because of the lack of tool for their construction. This point is particularly limiting as it makes it impossible to apply the method for systems whose size models is more important.

Tables III and IV expose sizes of different models obtained during the application. For the first case, it is not possible to compare the size of final models since Fault-hiding method cannot be used. However, for the second case, the Reconfiguration Block model is slightly twice bigger in term of states and transitions than the Reconfigurator model.

The fault-hiding method [4] allows a fault-tolerant control without a diagnoser. However, it is not applicable to all types of faults, and requires to model the faults and their impact on the system. On the other hand, the reconfiguration control method [5] is applicable to a larger number of errors, but requires the use of a diagnoser to detect the occurrence of faults. In addition, there is no tool facilitating the application of the method.

REFERENCES

[1] Blanke M., Kinnaert M., Lunze J. et Staroswieci M. *Diagnosis and Fault-Tolerant Control.* Springer, 2006.

[2] Wen Q. and Kumar R. A framework for fault-tolerant control of discrete event systems. *IEEE Transactions on Automatic Control*, 53(8), pp. 1839-1849, September 2008.

[3] Ru Y. and Hadjicostis C. Fault-tolerant supervisory control of discrete event systems modeled by bounded petri nets. American Control Conference, ACC'07, pp. 4945-4950 New York, 9-13 July 2007.

[4] Wittmann T., Richter J. et Moor T. Fault-hiding control reconfiguration for a class of discrete event systems. 4th IFAC Workshop on Dependable Control of Discrete Systems, DCDS2013, pp. 49-54, York, Royaume-Uni, 4-6 Septembre 2013.

[5] Nke Y. et Lunze J. Control reconfiguration based on unfolding of input/output automata. 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS2012, pp. 866-873, Mexico, Mexique, 29-34 August 2012.

[6] Paoli A., Sartini M. and Lafortune S. Active fault tolerant control of discrete event systems using online diagnostics. *Automatica*, 47(4), pp. 639-649, April 2011.