Fault tolerant control for manufacturing discrete systems by filter and diagnoser interactions

Alexandre Philippot, Pascale Marangé, François Gellot, Jean-François Pétin, Bernard Riera

Fault Tolerant Control for manufacturing discrete systems by filter and diagnoser interactions

A. Philippot[1], P. Marangé[2], F. Gellot[1], J.F. Pétin[2] and B. Riera[1]

[1]*Centre de Recherche en STIC (CReSTIC) – University of Reims Champagne-Ardenne (URCA), Reims, France*
{alexandre.philippot, francois.gellot, bernard.riera}@univ-reims.fr

[2]*Centre de Recherche en Automatique de Nancy (CRAN) – University of Lorraine, CNRS, Vandœuvre-lès-Nancy, France*
{pascale.marange, jean-francois.petin}@univ-lorraine.fr

ABSTRACT

The paper deals with an online safety mechanism to define interactions between a diagnoser and a control filter for fault tolerant control of manufacturing discrete systems. The diagnoser observes the plant behavior whereas the control filter ensures the safety from the controller. This online interaction is based by events communication where the control law is never reconfigured. The proposed approach is applied to CISPI platform from the CRAN laboratory (Research Center for Automatic Control of Nancy).

1. INTRODUCTION

Engineering systems become more and more complex and consequently, faults are more and more present and cause undesired behaviors. Diagnosis information can lead the user in its decision for maintenance or reconfiguration (Nke and Lunze, 2011), but can also allow fault tolerant control. The aim of diagnosis approaches is to detected and isolated with certainty a fault. After this step, it is necessary to reconfigure the controller in order to guarantee the dependability and safety but also to propose a Fault Tolerant Control (FTC) in a degraded mode (Blanke et al., 2003, (Paoli et al., 2011, Brown and Vachtsevanos, 2011).

Ensuring safety of manufacturing system control is currently based on two complementary approaches: control design activities with the objective to avoid unexpected behaviors and safe design activities by the development of online barriers.

First one, we focus on the control design activities with the objective to avoid unexpected behavior. Two main approaches are suggested in this way (Faure and Lesage, 2001): (i) control validation and verification (V&V) (Roussel and Faure, 2002), (ii) Supervisory Control Theory (SCT) based on synthesis controller (Ramadge and Wonham, 1989), that enables automatic generation of the controller from the specification, and the uncontrolled behavior of the plant. Most of the time, those designing approaches make two strong assumptions: the behavior of plant devices is not faulty and the designed control is exactly the same as the program that is implemented on the control devices (i.e. code generation deviations or code modifications by maintenance agents are not considered).

These assumptions being not realistic in practice, a second approach complements the safe design activities by the development of online barriers like diagnosis or filtering control. Diagnosis of manufacturing systems aims at detecting unsafe behavior of the plant and localizing the components that are involved in the behavioral deviation (Sampath, 1995). Control filtering aims at avoiding that a PLC program provokes plant damages, whatever the PLC program (Marangé, 2008, Riera et al., 2012). The filter is placed between the controller and the plant and inhibits potential dangerous evolutions by checking a set of safety constraints. Nevertheless, the diagnosis and the filter are formally built from models of process behavior. Consequently, hypothesis that the information from the process is correct is made. At least, if the plant situation is unknown, automatic procedures implemented by control filtering and diagnosis may be not efficient. This case generally requires the intervention of human expert to analyze the unknown situation of the plant, and to take emergency decision to drive back the plant in acceptable states.

The aim of this paper is to propose an approach of FTC where diagnosis provides information about the plant to the filter; and vice-versa. Control laws are never reconfigured but the system must always be in safety situation thanks to the filter even in case of plant fault. Models of the plant devices behavior as well as the control rules can be described as Discrete Event Systems (DES), i.e., dynamical systems with discrete state spaces and event-driven

transitions (Cassandras and Lafortune, 1999). The proposed approach provides similar results in term of detection to classical approaches (Sampath, 1995, Debouk, et al., 2000, Wang et al., 2007 …) but it continues to improve the safety even in presence of faults thanks to the control filter.

The paper is organized as follows. In section 2, the fault tolerant control architecture proposed is presented with a diagnosis and a filtering control sub-sections. A benchmark is studied with results in section 3 before to conclude and propose some future works.

2. FTC ARCHITECTURE

From the previous discussion, diagnosis approaches make hypothesis that controller information is safe whereas filtering controller approaches are supposed free of faults. The figure 1 presents the FTC architecture. Control law, diagnoser and filter are present in a Remote Terminal Unit (RTU) as a Programmable Logic Controller (PLC) for example. The diagnoser does not use directly the orders sent by the controller but the orders validated by the filter, which set to allows to guarantee the orders correctness. Also, the filter confirms orders according to the plant information (value of sensors/actuators) and the plant state defined by the diagnoser. User can send requests but also have situation awareness thanks to filter and diagnoser.

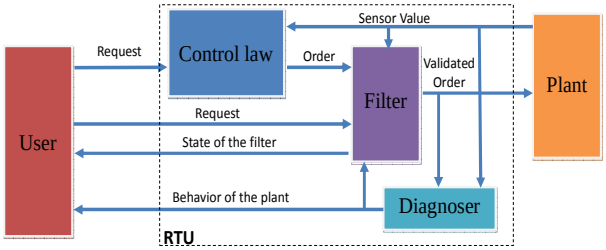

Figure 1. FTC Architecture

2.1. Diagnoser

In industrial processes, a manufacturing system is a functional chain composed of a controller that emits signals to a plant and receives sensor values. This exchange between controller and plant represents the only observable information available online. Since a diagnoser is defined as an observer of the system, it is necessary to use this information to rebuild behaviors through models.

From literature (Sampath, 1995, Qiu, 2005), centralized approaches appear as unthinkable for large and complex systems. As manufacturing system is composed of mechanical components (actuators/sensors), a methodology to obtain a decentralized diagnosis approach, as (Debouk, et al., 2000, Wang et al., 2007, Kan et al., 2010), for manufacturing systems with discrete sensors and actuators has been developed in previous works (Philippot and Carré-Ménétrier, 2011). It is composed of 4 offline steps describe:

1. From the plant components, decomposition is made to obtain local models called Plant Elements (PEs). A PE describes all possible mechanical evolution of the component independently of the controller.
2. From each PE, local desired behavior is extracted. Temporal information, obtained by excited events simulation, is added to enrich the model. The result is an automaton called Normal Behavior Model (NBM).
3. The third step identifies, from each normal state of NBMs, faults which can occur and composes the abnormal model by adding of labeled states to obtain local diagnosers (D_i). Faults are grouped according to the failing component (sensor/actuator) into partitions.
4. A High Level Diagnoser from global specifications is done for uncertainty cases.

Diagnosers are implemented as online observers in the PLC. User's decision is given thanks to the set of local labels.

A local diagnoser is a special case of an observer that carries fault information by means of labels attached to states. These labels indicate the types of faults that have been occurred. A local diagnoser is considered as an extended automaton: $D_i = (X_i \cup X_{DFi}, Z_{io}, \delta_i, x_{i0}, T_i, l_i)$ where:

- X_i is the set of normal states of NBM_i,
- X_{DFi} is the set of faulty states,
- Z_{io} is the set of observable events by the PE_i,
- δ_i: $X_i \times Z_i^* \rightarrow X_i \cup X_{DFi}$ is the transition function with the expected (δ_{ei}) and unexpected (δ_{ui}) functions from a state,
- x_{i0} is the initial state,
- T_i is the set of interval time where transition functions are expected between $[t_{min}, t_{max}]$,
- l_i is the set of decision functions of the local diagnoser D_i with $l_i(x)$ the decision function of the state x which can be one or more fault labels {Fj}. The sets of failure events corresponding to partitions, noted Π_f.

Indeed, the methodology is dependent of the control specification (step 2) and if the controller is not safe or if it changes, then diagnosers can return a bad decision in the first case or must be reconstructed in the second case. To have diagnosis independent from the control, diagnoser is obtained from the behavior of PE and the addition of the possible faulty events.

From decentralized diagnosers, a transition function δ_i corresponds to a logical expression composed by all the events. It is possible to define all transition functions by the 2^n possibility (with n: number of events and intervals). However, the mechanical structure of components and the use of filters make it impossible some combinations. For example, only one interval time can be activate simultaneously, or thanks to the control filter, opposite orders cannot be sent. Consequently, the complexity

depends on the granularity of the local models but also on the performance of the control filter. These diagnosers are independent of the controller specification in its structure thanks to the control filter but not in the definition of the set of interval T_i.

The choice of an automaton to represent a local diagnoser permits to compose a library of commonly components. However, this model can be translated as Markov chain or Causal Temporal Signature under some hypothesis.

2.2. Control Filter

The control filtering consists in interlacing a filter between the plant and the control law to inhibit the evolutions that can lead the system to a dangerous situation for operators and production resources. This aim is to ensure that the controller outputs (Σ_c), are legal according to plant safety. It means that, for each new evolution of actuators output vector (at *t*), the filter verifies that these outputs are compatible with the plant state perceived by means of uncontrollable variables Σ_{uc} (inputs sensors (at *t, t-1, t-2…*), previous outputs (at *t-1, t-2…*), observers (at *t, t-1, t-2…*)).

The filter is built according to a set of logical constraints that must be satisfied to let the outputs getting out of the control filter. It is based on the use of safety constraints, which act as logical guards placed at the end of the PLC program, and forbids sending unsafe controllable events to the plant (Marangé et al. 2008), (Riera et al., 2014). Constraints (or guards) are always modeled with the point of view of the control part (PLC), and it is assumed that the PLC scan time is sufficient to detect any changes of the input vector (synchronous operation, possible simultaneous changes of state of PLC inputs).

Safety constraints are expressed in the form of a logical monomial function (product of logical variables, as $\prod$) which must always be equal to 0 (FALSE) at each PLC scan time in order to guarantee the safety. It is considered in this work that the initial safe state for all the actuators (o_k) is defined to 0.

Initially, the constraints are defined in order to ensure a permissive control, and it is assumed that, with the filter, the system remains controllable. In other words, it is possible to design a controller which matches the specifications. For example, considering the previous hypothesis about the safe initial state, a filter which resets all outputs is safe but does not ensure the controllability. Some guards involve a single output at time *t* (simple safety constraints *CSs*), other constraints involve several outputs at time *t* (combined safety constraints *CSc*). Constraints require the knowledge of Σ_c and Σ_{uc} at the current time *t* and possibly previous times (presence of edge (*t-1*) for instance noted *). Hence, the filter requires a memory function.

The set of constraints CS is considered as necessary and sufficient to guarantee the safety. In this approach, it is assumed that safety constraints can always be represented as a monomial and depend on the uncontrollable and controllable variables (at *t, t-1, t-2…*). Filter stops has to stop the process in a safe situation if a safety constraint is not respected.

CSs and *CSc* can be represented respectively by equation (1) and equation (2) which are Boolean monomial functions and have always to be False at each PLC scan time. N_{CSs} and N_{CSc} are respectively the number of simple safety constraints and the number of combined safety constraints. N_o is the number of outputs.

$$\forall m \in [1, \mathrm{N_{CSs}}],\ \exists!\, k \in [1, \mathrm{N_O}]\ /\ CSs_m = \prod(o_k, \Sigma_{uc}) = 0 \quad (1)$$

$$\forall n \in [1, \mathrm{N_{CSc}}],\ \exists!\, (k, l, \dots) \in [1, \mathrm{N_O}]\ with\ k \neq l \neq \cdots\ /\ CSc_n = \prod(o_k, o_l, \dots, \Sigma_{uc}) = 0 \quad (2)$$

There are 2 forms of Simple Safety Constraints *CSs* because they are expressed as a monomial function, and they only involve a single output at time *t* (equation (3) or (4)):

$$\forall m \in [1, \mathrm{N_{CSs}}],\quad \exists!\, k \in [1, \mathrm{N_O}]\ /\ CSs_m = o_k . h_{0m}(\Sigma_{uc}) \quad (3)$$

xor

$$CSs_m = \overline{o_k} . h_{1m}(\Sigma_{uc}) \quad (4)$$

These simple safety constraints (*CSs*) express the fact that if $h_{0m}(\Sigma_{uc})$ which is a monomial (product) function of only uncontrollable variables at *t*, is TRUE, o_k must be necessarily FALSE (equation (3)) in order to keep the constraints equal to 0. If $h_{1m}(\Sigma_{uc})$ is TRUE, o_k must be necessarily TRUE (equation (4)).

For each output, it is possible to write equation (5) corresponding to a logical OR of all simple safety constraints.

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(f_{sk}(o_k, \Sigma_{uc}) \right) = 0 \quad (5)$$

$f_{sk}(o_k, \Sigma_{uc})$ is a logical $\sum\prod$ function independent of the other outputs at *t* because only *CSs* are considered. $f_{sk}(o_k, \Sigma_{uc})$ can be developed in equation (6) where f_{s0k} and f_{s1k} are polynomial functions (sum of products, $\sum\prod$) of uncontrollable variables. Equation (6) has always to be FALSE because all simple safety constraints must be FALSE at each PLC scan time.

$$f_{sk}(o_k, \Sigma_{uc}) = o_k . f_{s0k}(\Sigma_{uc}) + \overline{o_k} . f_{s1k}(\Sigma_{uc}) = 0 \quad (6)$$

Taking into account all *CSs*; it is possible to write equation (7).

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(o_k . f_{s0k}(\Sigma_{uc}) + \overline{o_k} . f_{s1k}(\mathrm{I}, \Sigma_{uc}) \right) = 0 \quad (7)$$

The definition of constraints set is not formal and the filter robustness must be verified. In (Marangé, 2008) and (Riera et al., 2012), authors proposed to enrich this expert-based approach by a formal identification of the constraints set to ensure its completeness.

The use of this filter allows detecting errors resulting from the controller by making a hypothesis on the accuracy of the information resulting from the plant. Indeed, a fault on the plant can lead:

- Too much restriction: sensor information is going to be blocked in the most critical state and the constraint is not verified while the plant is not in a critical situation.
- Too much tolerant: sensor information is going to be in the state which verifies all the time the constraint and thus the filter is going to allow to pass dangerous orders for the plant. This case is to be avoided.

The consideration of diagnosis information allows to use the filter in degraded mode. For that purpose, the information resulting from the plant is added by taking into account diagnoser. When a failure arises on a sensor or an actuator, the filter constraints that contain the logical variables associated to the faulty devices becomes unreliable. Authorized signals may be forbidden, and, worse forbidden signals may be authorized. Consequently, the filter constraints must consider the occurrence of a fault or not.

For every fault partition, a flag is set to true when the diagnoser reaches a faulty decision state. This flag determines if the considered variable can be used into the filter constraint (flag=0), or if an equivalent reconstructed information must be used (flag=1). Only the sensor information can be reconstituted by using:

- the expert knowledge (timed or temporal model),
- redundant information or reconstruction logics.

The property defining the dangerous situation has been verified using a model-checker meaning that the filter delivers correct inhibition and authorization even in presence of device faults (with the assumption that the diagnoser is able to detect and localize the fault).

Moreover, as the control filter only concerns safety part and not the functional part, if the component is exchanged or replaced, only the set of constraints corresponding to this component must evolve. For industrial systems, establishment of a constraints library is feasible. In fact, constraints sets are defined for a sub-system of component interaction.

3. CASE STUDY

The approach is applied to the CISPI platform from the CRAN laboratory (figure 2). This platform implements hydraulic processes involving valves, pumps and tanks and various transmitters (flow, pressure…). Local controllers implement basic control loops and are involved in a global mode management control that enables concurrent access to devices for start, shutdown and normal operation procedures. To avoid damages and failures of the system, as well as the human operator's errors, this experimental platform promotes new forms of control organization that exploits the capacity ambient technologies (sensor network, PDA, mobile control...) to favor safe human/system interactions in any place, at any instant and for any plant operation.

Within the framework of this project, the control filter and diagnoser are implanted to bring a help during the supervisory control of the CISPI system. To illustrate the approach presented in this paper, an automatic valve is considerate. This valve can be closed or open by respectively *C* and *O* boolean signals, and two sensors for the open position (*fso*) and for the closed position (*fsc*) are present.

Independently of the control laws, the sub-system valve must always be in a safety mode. For this, an assumption is made that when a fault is on an actuator, all outputs must be deactivated by the filter. If a fault is on a sensor, the sub-system can be tolerant to this fault.

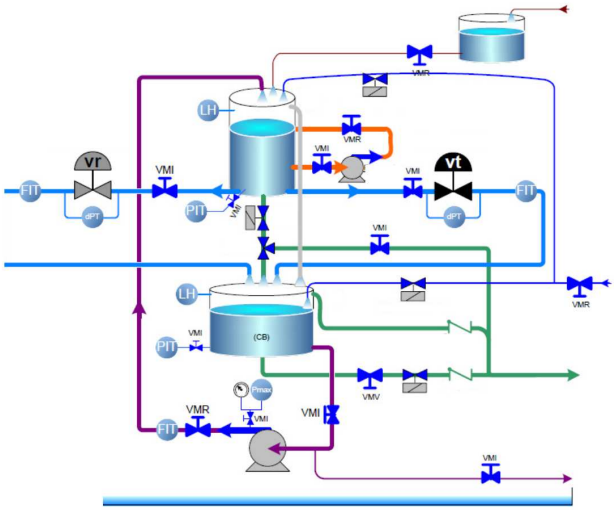

Figure 2. CISPI Platform

3.1. Diagnoser

From the illustrative example, the valve with sensors *fsc* and *fso* constitute one PE and it is possible to identify each faulty event by a label:

- Sensor *fsc* stuck to 0 (F1) or to 1 (F2)
- Sensor *fso* stuck to 0 (F3) or to 1 (F4)
- Valve stuck to *fsc* (F5) or *fso* (F6) position
- Unexpected *fsc* (F7) or *fso* (F9) from 0 to 1
- Unexpected *fsc* (F8) or *fso* (F10) from 1 to 0
- Unexpected movement from *fsc* to *fso* (F11) or from *fso* to *fsc* (F12)
- Valve blocked between *fsc* and *fso* (F13)

Three fault partitions are defined belong to:

- Sensor *fsc*: Π_{fsc} = {F1, F2, F7, F8}
- Sensor *fso*: Π_{fso} = {F3, F4, F9, F10}
- Valve: Π_{Va} = {F5, F6, F11, F12, F13}

With the consideration of the controller information, and thanks to the filter, the valve diagnoser is composed of 9 normal states and 16 abnormal states (Fig. 3) where:

- double circle is the initial state,
- 9 white states are the normal states,
- 3 grey states noted F2, F7, F8 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{fsc},
- 3 grey states noted F4, F9, F10 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{fso},
- 4 grey states noted F5, F6, F11, F13 represent the abnormal states with detection and isolation of an abnormal behavior with certainty from Π_{Va},
- 6 black states describe the detection of a fault but not the isolation (4 intermediate before isolation).

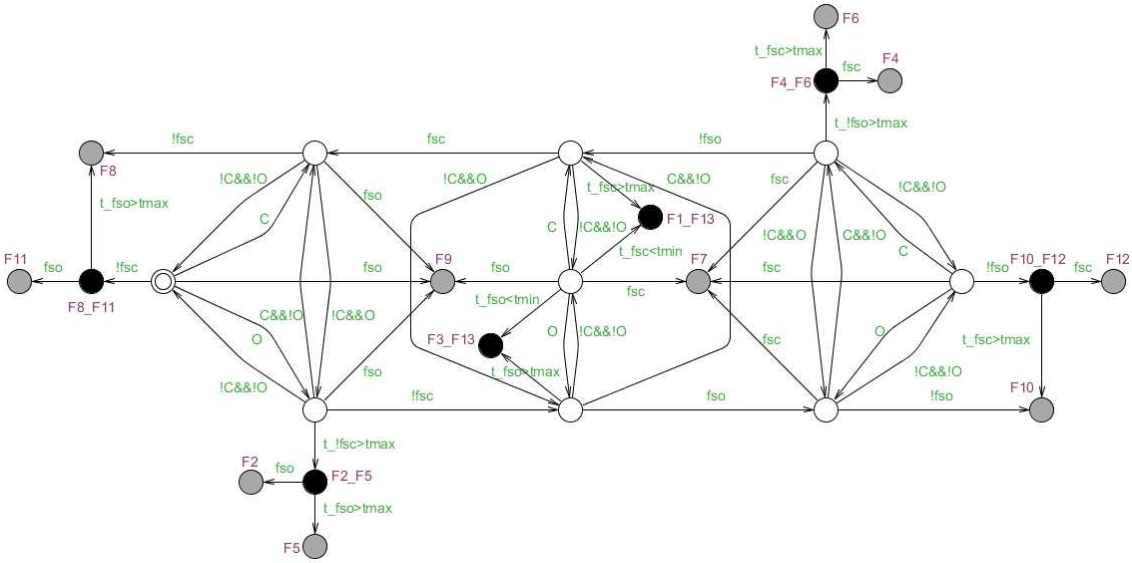

Figure 3. Valve Diagnoser

The reliability of sensors ensures to be into a safety mode (white states). However, after the detection and isolation of a fault (grey and black states), this diagnoser cannot be anew used. Indeed, it is not possible to rely on misinformation. That is why, it is necessary to preserve the state of the system until the fault is been corrected and reset.

3.2. Control Filter

Constraints take into account information of the diagnosers. Information used in the filter is noted X_{filter} and diagnosis information is noted *defX*. The following flags are done:

- *deffso* for the partition of valve sensor *fso*,
- *deffsc* for the partition of valve sensors *fsc*,
- *defV* for the partition of valve actuator *V*,

To be tolerant on sensors' faults, an expert knowledge is used to estimate the plant information by temporal information. This knowledge can be optimally obtained by FMEA (Failure Mode and Effects Analysis) and so provide a reactivity of detection. For example, figure 4 shows equivalent information of *fso* and *fsc* sensors information from a learning chronogram where the estimated value of *fso* is given by a flag *TON1* when an On Delay Timer is activated, and respectively a flag *TON2* for the estimated value of *fsc*.

- for $fso = 1$ by $\widehat{fso} = TON1$
- for $fsc = 1$ by $\widehat{fsc} = TON2$

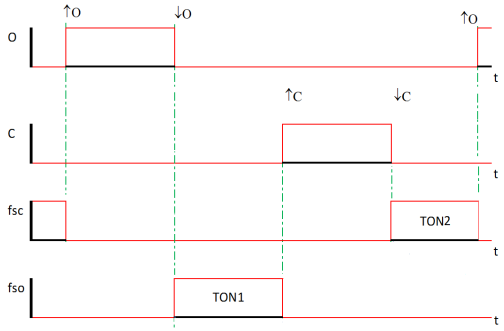

Figure 4. Reconstruction of sensors' information

For a sensor fault, the plant information is replaced by temporal information:

$$fso_{filter} = \overline{deffso}\,.fso + deffso.\widehat{fso} \qquad (7)$$

$$fsc_{filter} = \overline{deffsc}\,.fsc + deffsc.\widehat{fsc} \qquad (8)$$

No information can be estimated for outputs *C* and *O*. Consequently, orders must be deactivated by the filter even in case of faulty event by:

$$C_{filter} = \overline{defV}.C \quad (9)$$
$$O_{filter} = \overline{defV}.O \quad (10)$$

The set of constraints is defined as following. It is forbidden to maintain an order when the position valve is done (equations (11) & (12)). It is forbidden to deactivate an order until the ending position valve is not done (equations (13) & (14)). It is forbidden to activate an order until the starting position valve is not done (equations (15) & (16)). For the combined safety constraint of equation (17), it is forbidden to activate orders *C* and *O* together:

$$CSs_1 = C_{filter}.fsc_{filter} = 0 \quad (11)$$
$$CSs_2 = O_{filter}.fso_{filter} = 0 \quad (12)$$
$$CSs_3 = C^*_{filter}.\overline{C_{filter}}.\overline{fsc_{filter}} = 0 \quad (13)$$
$$CSs_4 = O^*_{filter}.\overline{O_{filter}}.\overline{fso_{filter}} = 0 \quad (14)$$
$$CSs_5 = \overline{C^*_{filter}}.C_{filter}.\overline{fso_{filter}} = 0 \quad (15)$$
$$CSs_6 = \overline{O^*_{filter}}.O_{filter}.\overline{fsc_{filter}} = 0 \quad (16)$$
$$CSc_1 = C_{filter}.O_{filter} = 0 \quad (17)$$

Where $\overline{X^*}.X$ and $X^*.\bar{X}$ represent respectively a rising and a falling edge of an order *X*.

$$CSs_3 = C^*_{filter}.\overline{C_{filter}}.\overline{fsc_{filter}} = 0 \quad (13)$$
$$CSs_4 = O^*_{filter}.\overline{O_{filter}}.\overline{fso_{filter}} = 0 \quad (14)$$

3.3. Results and Key Performance Indicators

A first analysis shows that the system is detectable in a bounded delay with certainty for the defined fault partitions. Indeed, all labels are represented in an abnormal state. However, the system is non-diagnosable with certainty. 10 labels on 13 possible are isolated with certainty (one unique label), 3 labels are with an ambiguity. For example, it is not possible to isolate with certainty states with labels {F1, F13} and {F3, F13}. Diagnostic Coverage (DC) is the ratio of the probability of detected dangerous failures (dd) to the probability all the dangerous failures (d). This meaning of the term DC is common to (ISO13849-1) and (IEC/EN 62061). For the valve, the DC is to 76.9%. The standard ISO13849-1 divides DC into four basic ranges: i) <60% = none, ii) 60% to <90% = low, iii) 90% to <99% = medium and iv) 99%+ = high. Consequently, another rule must be present to improve it and to guarantee complete diagnosability notion as defined in (Lin, 1994).

Table 1 presents a comparison between solutions with or without filter and/or diagnosers by simulation of the 13 faulty events under ProcesSim (http://processim.hecfh.be/). Thirteen scenarii have been exploited to obtain these results. With no filter, the valve system is under blocked behavior in 8 cases, into a degraded mode in 1 case and induces a defect situation for 4 cases. We can see that the tolerant situation disappear with the use of the filter only because its purpose is to ensure a safety behavior. When the FTC solution is used, the degraded mode is tolerant to 4 faulty events and above all, it decreases 2 cases of defect situations.

The proposed FTC approach has not been extended on all CISPI platforms yet. But a study has been done on a sub-system composed of 2 automatic valves, one pump and 2 tanks. Another point of view can be also to evaluate the steady state transition probabilities as a KPI. Indeed, a repetitive sequence of normal events can provide an indicator of the system behavior. For the moment, this remark is not treated in these works.

Table 1: Comparison with and without FTC solution

	Diag No Filter	No Diag Filter	FTC (Diag and Filter)
Blocked	8	9	7
Tolerant	1	0	4
Defect	4	4	2

4. CONCLUSION

A Fault Tolerant Control approach is presented around an interaction between diagnosers and filtering control. Diagnosis design is refined using enriched information from the real implemented control rules (control + violated constraints of the filter) while control filter benefits from using diagnose information to adapt its set of constraints according to reliable raw or constructed information.

In future works, when diagnosers detect a fault on a component or when the filter detects a mistake on the controller, a significant explanation must be given to a human operator to choose the best policy. A graduated explanation with potential consequences is to return. As last remark, the control filter has been implemented and extended to control design pattern on a real complex system called *CellFlex* at the University of Reims (www.univ-reims.fr/meserp/).

REFERENCES

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2003). Diagnosis and Fault-Tolerant Control. *Springer-Verlag*.

Brown D.W. and Vachtsevanos G.J. (2011). A Prognostic Health Management Based Framework for Fault-Tolerant Control. *Annual Conference of the Prognostics and Health Management Society* (PHM'11), Montreal, Quebec, Canada.

Cassandras C.G., Lafortune S. (1999). Introduction to discrete event systems. *Kluwer Academic Publishers*, Dordrecht.

Debouk R., Lafortune S. et Teneketzis D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete events systems. *In Journal of Discrete Event Dynamical System : Theory and Application*. pp.33-86.

Faure J-M., Lesage J-J. (2001). Methods for safe control systems design and implementations. *10th IFAC*

Symposium on Information Control Problems in Manufacturing, INCOM'2001, Vienna, Austria.

IEC/EN 62061. Safety of machinery: Functional safety of electrical, electronic and programmable electronic control systems. (2005).

ISO13849-1. Safety of machinery. Safety-related parts of control systems. General principles for design. (2006).

Kan John P., Grastien A. and Pencolé Y. (2010). Synthesis of a Distributed and Accurate Diagnoser. *21st International Workshop on the Principles of Diagnosis* (DX'10), Portland, Oregon, USA.

Lin F. (1994). Diagnosability of Discrete Event Systems and its Applications. *In Discrete Event Dynamic Systems*, 4, *Kluwer Academic Publishers*, Boston, USA.

Marangé P. (2008). Synthèse et filtrage robuste de la commande pour des systèmes manufacturiers surs de fonctionnement. PhD of the University of Reims Champagne-Ardenne.

Nke Y., Lunze J. (2011). Online control reconfiguration for a faulty manufacturing process. *3rd International Workshop on Dependable Control of Discrete Systems* (DCDS'11), Saarbrücken, Germany.

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

Philippot A. and Carré-Ménétrier. V. (2011). Methodology to obtain local discrete diagnosers. *3rd International Workshop on Dependable Control of Discrete Systems* (DCDS'11), Saarbrücken, Germany.

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

Ramadge G., Wonham W. M. (1989). The control of discrete event systems, *Proc. IEEE, Special issue on DEDSs*, 77, pp.81-98.

Riera B., Annebicque D., Gellot F., Philippot A., Benlorhfar R. (2012). Control synthesis based on logical constraints for safe manufacturing systems. *14th IFAC Symposium on INformation COntrol problems in Manufacturing* (INCOM 2012), Bucarest, Romania.

Riera B., Coupat R., Philippot A., Gellot F. and Annebicque D. (2014). Control design pattern based on safety Boolean guards for manufacturing systems: application to a palletizer. *12th IFAC-IEEE International Workshop On Discrete Event Systems* (WODES'14), France.

Roussel J.M. and Faure J.M. (2002). An algebraic approach for PLC programs verification. *In Proceedings of 6th international Workshop On Discrete Event Systems*, Zaragoza, Spain, pp.303–308.

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

Wang, Y., Yoo, T. S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems*, Vol.17(2), pp233-263.

BIOGRAPHIES

Philippot Alexandre was born in France in 1979. After a Master degree in Systems Optimization and Safety at the University of Reims Champagne-Ardenne in France, he achieved a PhD in Diagnosis of Discrete-events Systems at the University of Reims Champagne-Ardenne in France in 2006. He passed a PostDoc at the Ecole Normale Supérieure de Cachan (ENS-Cachan) in France (2007). Research interests: Discrete Event Systems, Fault diagnosis, Modeling, Supervisory Control Theory, Optimal Control, Manufacturing systems. Currently, he is associated professor at the University of Reims Champagne-Ardenne (URCA) and realises its research at CReSTIC Laboratory (research center in Sciences and Technologies on Information and Communication).

Pascale Marangé was born in 1982 in France. After a Master degree at the University of Reims Champagne-Ardenne (URCA) in France, she achieved a PhD in Control synthesis of Discrete-events Systems at URCA in 2008. Currently, she is associated professor at Lorraine University and realizes its research at Nancy Research Center for Automatic Control (CRAN). Her research interests include Verification and Validation of PLC program, Dependability of Discrete-Event Systems, Control synthesis.

François Gellot is associated professor at the University of Reims Champagne-Ardenne (URCA) France. He was born in France in 1965. He achieved a PhD in analyze and simulation of Petri nets (1994). Its research interests include supervisory control, discrete events systems modeling and Verification and Validation of PLC program.

Jean-François Pétin is a Professor at Lorraine University and realizes its research at Nancy Research Center for Automatic Control (CRAN). Its research interest concerns Dependability of Discrete-Event Systems, Control synthesis, Verification & Validation.

Bernard Riera received the Ph.D. degree in Automatic Control from the University of Valenciennes (UVHC), in 1993. He is a Professor of Control Engineering at the University of Reims Champagne-Ardenne (URCA) France, a Researcher at the CReSTIC (research center in Sciences and Technologies on Information and Communication) and associate director of the CReSTIC. Its research interests include supervisory control, supervisory support systems, discrete events and hybrid systems modeling.