DETECTABILITY AND DIAGNOSABILITY OF DISCRETE EVENT SYSTEMS

Application on manufacturing systems

Moamar Sayed Mouchaweh, Alexandre Phillipot, Véronique Carré Ménétrier
Université de Reims, CReSTIC - LAM, Moulin de la Housse B.P. 1039, 51687 REIMS Cedex 2, FRANCE.

Keywords: Discrete Event Systems, Modelling, Diagnosis, Manufacturing systems.

Abstract: The diagnosis is defined as the process of detecting an abnormality in the system behavior and isolating its causes or sources. Not all the systems are diagnosable. Thus, before Appling a method to diagnose a system, we need to know if this system is diagnosable according to the set of failures required to be detected and isolated. This paper presents an algorithm to determine if a system is detectable or not, i.e., if we can know, at each instant, whether the system works under a normal or abnormal functioning state. In the case that the system is detectable, this algorithm determines if this system is diagnosable. This algorithm combines event and state based approaches in order to maximise the diagnosability power with a minimum number of sensors. In addition, the time is integrated and modelled with fuzzy intervals to enhance this diagnosabilty power and to take into account the imprecision of events occurrences instants. An example of manufacturing system is used to illustrate the functioning of this algorithm.

1 INTRODUCTION

The complexity of industrial systems increases rapidly while in the same time safety, availability, reliability, and performances of these systems rise. Consequently, the potential for system to fail is enhanced regardless how safe the designs are, how improved the quality of control techniques are and how better trained the operators are (Perrow, 1984).
Discrete Event System (DES) is dynamic systems equipped with a state space and a state-transition structure. One of the key benefits of a DES is that there is no need to descretize time and yet one can capture the asynchronous nature of event processes. DES is often modelled in using a finite-state automaton, a Petri net or process algebra. Each modelling tool has its advantages and disadvantages depending on the objectives of modelling: model complexity or natural projection and formalisation facilities.
When failures occur in a system, observations are analysed using the system model to generate a set of possible failures. A failure implicates one or more of system components and explains all the observed measurements: deviating and normal. Generally, failures can be characterized as: permanent or intermittent. The permanent failures can be divided into two major kinds: progressive or abrupt. The type of failures can be classified as: sensor failures, actuator failures, process failures and control loop or controller failures.
In this paper, a quick review of major approaches of failure diagnosis of DES is presented. Since not every system is diagnosable, these approaches define a notion of diagnosability based either on event, on state or both depending on their model and the type of failures: permanent or intermittent, that they must detect and diagnosed. Then a method to diagnose DES is introduced. This method is modular and decentralized. Before applying this method to diagnose DES, an algorithm is used to determine if the process is diagnosable according to the observable events and for the set of failures to be isolated. If the system cannot be diagnosed, then sensor maps must be modified in order to provide the diagnosability property to the system. Finally, an example of a manufacturing system is used to illustrate the diagnosis method with the diagnosability algorithm.

Sayed Mouchaweh M., Phillipot A. and Carré Ménétrier V. (2005).
DETECTABILITY AND DIAGNOSABILITY OF DISCRETE EVENT SYSTEMS - Application on manufacturing systems.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Signal Processing, Systems Modeling and Control*, pages 149-154
DOI: 10.5220/0001176201490154

2 FAILURE DIAGNOSIS APPROACHES

There is a great deal of methods for designing and developing an automated diagnosis system (Ramdage, 1987), (Su, 2004), (Wang, 2000), (Cassandras, 1999). A common feature of these methods is the use of a model to specify the correct behaviour of the system and then to analyze the observations of the system current operating state to detect a failure. The choice of one of these methods depends on several factors as: the dynamic of the system (discrete, continuous and hybrid), implementation standpoint (on line, off line), information representation (quantitative, qualitative), system complexity (large or simple) and the depth of available information about the system specifications and behaviour (structural, analytical and heuristic knowledge).
We will focus on the DES failure methods. They can be classified according to the structure of their plant and diagnoser models into 3 main categories: centralized, decentralized and distributed structures.

3 STRUCTURE OF DES FAILURE DIAGNOSIS METHODS

There are three main structures of plant and diagnoser models of DES:

1. Centralized approaches: There is one centralized system model associated with one centralized diagnoser, which collects observations, then makes a final decision about the target system's fault status. In (Zad, 1998), we can find an example of these approaches.
2. Decentralized approaches: There is one centralized system model associated with several local diagnosers, each of which receives observations from a specific set of sensors and makes local diagnosis decision based on such local observations. A very limited communication is permitted through a centralized coordinator to solve the problem of the possible ambiguity between local diagnoser decisions. In (Sampath, 1994), an example belongs to this category.
3. Distributed approaches: The system consists of several local components, and is associated with several local diagnosers, each of which is usually responsible for a specific local component. Since neither a centralized system model nor a centralized coordinator exists, a pure concurrent communication among local diagnosers is necessary. In (Su, 2004), (Holloway, 1994), we can find an example belonging to this category.

4 DES DIAGNOSABILITY NOTIONS

The DES failures diagnosis methods can be divided into two main categories: event-based methods and state-based methods. In event-based methods, failures are modelled as execution of certain faulty events. The DES plant representation is based on a finite-state automaton. This model accounts for the normal and failed behaviour of the system. All information relevant to the diagnosis including sensors information is captured in the event set of the model. Typically, the observable events in the system are one of the following: commands issued by the supervisor and sensor readings immediately after the execution of the above command, and changes of sensor readings. The unobservable events are failure events or other events which cause changes in the system state not recorded by sensors. This model is obtained by a product composition of finite-state machines models of individual system components.
A diagnoser is designed to decide whether the original behaviour contained a fault or not in basing on sequences of observable events. The diagnoser should announce a fault at most n steps after the fault occurred. Once a fault is announced, the diagnoser cannot stop announcing it (Sampath, 1994).
To enhance the diagnosability, the above framework is extended to dense-time automata (Tripakis, 2001). This extension is useful since it permits to model plants with timed behaviour. It also allows diagnosers to base their decisions not only on the sequences of observed events, but also on time delays between these events.
An event-based method is proposed in (Garcia, 2003) for monitoring and diagnosis of manufacturing systems. To detect a special event (failure), a monitoring observer (agent) analyzes discrete event signals triggered by entities as they transit through the monitored system. In (Holloway, 1994), (Deepa, 2000) the authors present an approach to fault monitoring in manufacturing systems allowing the modelling of process in which both single-instance and multiple-instance behaviours are exhibited concurrently. The timed sequence events generated by the DES under supervision is compared with a set of specifications

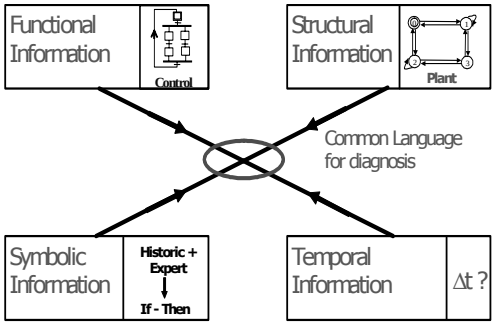

Figure 1: Information sources

of normal functioning called templates. These templates are suitable for manufacturing systems and can operate independently in parallel.

An equivalent state-based approach is considered where the occurrence of a failure is modelled as reaching of certain faulty states (Lin, 1994). This approach uses a general model for both types of diagnosis: off-line and on-line. For on-line, a deterministic Moore automaton with partial state observation and no event observation is used while in off-line diagnosis a nondeterministic Mealy automaton with no state observation and partial event observation.

In (Zad, 2003) a based state model to realize the passive diagnosis was proposed. The state set of system is partitioned according to system condition (failure status). A diagnoser, based on finite state automata, estimates the conditions in using the sequence events and state outputs.

In (Philippot, 2005), we have proposed a method to realize the diagnosis of DES. In the next, we define an algorithm to determine if the system is diagnosable, this algorithm is adapted for our method and it limits the number of states necessary to detect actuator-sensor failures as we will see later.

5 PROPOSED APPROACH TO FAILURE DIAGNOSIS

In (Philippot, 2005), a model-based approach to diagnose DES is proposed. A mathematical model G must be constructed to define how system states change due to event occurrences. The model is decentralized: the system consists of several local components (G^i, $i = 1 .. n$) with a coordinator to realize a minimum of communication between these local components. The diagnoser model is distributed: several local diagnoser (D^i, i = 1 .. n) are constructed, each one of them is usually responsible for a specific local component. The goal of the use of decentralized model and a distributed diagnosis is to reduce the spatial explosion problem at the design stage and to facilitate the localisation of default elements.

A notion of diagnosability is defined since not every system is diagnosable. This notion depends on the partition required (failures to isolate) and on the observable events.

To enhance the diagnosability, time is integrated to this approach. Fuzzy functions, modelling the minimum and maximum expectation moments accepted for an event to take place, are computed. The use of fuzzy intervals is useful to better take into account the imprecision and uncertainty attached to the calculation of these moments and to better model the tractability which can be used to realize a prognostic, particularly useful for the progressive failures.

This approach uses different representation tools (automata, rules, algebraic and mathematical equations, ..) according to the available information. The goal is to enrich the model in using all the available information sources with a suitable representation tool to be able to realize the diagnosis. These sources are (see figure 1):

- functional information contained in the process schedule conditions,
- structural information coming from the process itself and the sensors-actuators spatial distribution,
- symbolic information given by experts and/or previous experience obtained by a learning set of previous functioning,
- temporal information coming from the space and temporal parameters of process actuators and sensors.

For this approach, three models are defined: plant control and diagnoser models.

5.1 Plant model

A plant model is divided into several components. Each model G^i and corresponding language L_i describe the logical, untimed behaviour of the monitored system. $G = (M, \Sigma_c)$ where M is a Moore automaton: $M = (\Sigma, Q, Y, \delta, h)$: Σ is the set of finite events, Q is the set of states, Y is the output space, $\delta: \Sigma \times Q \rightarrow Q$ is the state transition function. $\delta(\sigma, q)$ gives the set of possible next states if σ occurs at q. $h: \Sigma \times Q \rightarrow Y$ is the output function. $H(\sigma, q)$ is the observed output when σ occurs at q. $\Sigma_c \subseteq \Sigma$ is the set of controllable events. $\Sigma_o \subseteq \Sigma$ is the set of observable events where $\Sigma_c \subset \Sigma_o$. An automaton is

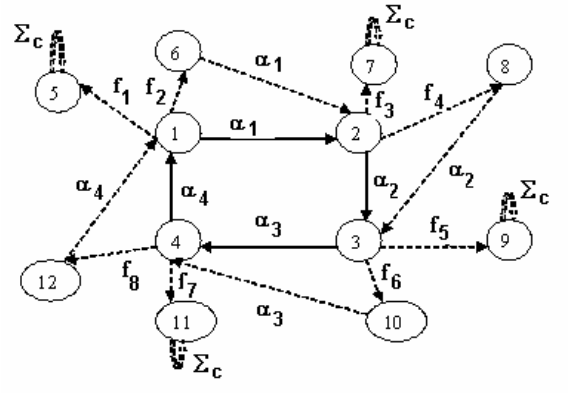

Figure 2: Functional and diagnoser models

used for each model. This automaton takes into account all the observable events. The detailed explication of the construction of this model can be found in (Philippot, 2004), (Philippot, 05).

5.2 Control model

The controller model is constructed in using a Sequential Function Charts. This letter reflects the functional information corresponding to the schedule conditions for all system components. The SFC is chosen to model the controller because it is well adapted for industrial applications especially manufacturing systems.

5.3 Diagnoser model

The set of failures to be detected and isolated must be defined as well as the type of these failures. Let $\sum_f = \{f_1, f_2, .., f_n\}$ be the set of failures to be detected and $\Pi = \{S_N, S_{f_1}, S_{f_2}, .., S_{f_n}\}$ denotes the set of normal partition and the type of failures to be isolated: sensor or actuator and which sensor or actuator. Additionally, the nature of failure must be defined: permanent or non-permanent.

Before constructing the diagnosers, a notion of diagnosability must be defined to determine if the system is diagnosable or not. A system is detectable according to a set of observable events and a set of partition iff each normal state can be distinguished from all the failures:

$$H(S_N) \neq H(S_{AN}) \Rightarrow \\ \forall q \in S_N, \forall q' \in S_{AN} \Rightarrow h(q) \neq h(q') \quad (1)$$

Then, the system is diagnosable if we can distinguish each partition of the set Π. This isolation must be determined within a bounded number of events and a bounded time:

$$\forall q, q' \in S_{AN} : h(q) \neq h(q') \quad (2)$$

Let us take the following example (see figure 2) of a process which has 4 normal states S_N = {1, 2, 3, 4} with the observable events $\Sigma_o = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. The partition to be identified is S_{AN} = {5, 6, 7, 8, 9,10, 11, 12} which correspond to the failure events $\sum_f \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$.

The outputs of each state, thank for the sensors are: $H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}, h_{12}\}$. The figure 2 shows the functional model corresponding to the product of the model plant with the control model, in solid lines, and the diagnoser model in dotted lines.

The failure events: f_1, f_3, f_5 and f_7, indicate an equipment failures, actuator, as stuck on or stuck off. These events are non-observables and must be detected either by an observable event which entails a state with an output different from the estimated one, in the case of stuck on, or by the fuzzy functions of expected events occurrence instants in the case of stuck off. The other failure events denote the sensor failures. These events can be observable or non-observable. In the case of observable ones, the failure is detected at once without any delay. This fact is important for the functioning of the SFC in order to permit the evolution of the command. Indeed our approach requires a validate sensor values before permitting a new command. In the case of non-observable events, as in the figure 2, the failure must be detected in using only the sensors outputs. In this case the non-observable failure events correspond to a sensor stuck on, level 1, or stuck off, level 0. The number of sensors must be enough to distinguish all the states of normal and abnormal states.

We can find that our approach combines event and state based approaches in order to maximise the diagnosability power with a minimum number of sensors. In addition the time is integrated and modelled with fuzzy intervals to enhance this diagnosabilty power and to take into account the imprecision and uncertainty of time occurrence events.

Let us explain our algorithm to know if the system represented by the figure 2 and for $\Sigma_o = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ and S_{AN} = {5, 6, 7, 8, 9, 10, 11, 12} is diagnosable. The first step of our algorithm is to construct a matrix $M(nxm)$ where n is

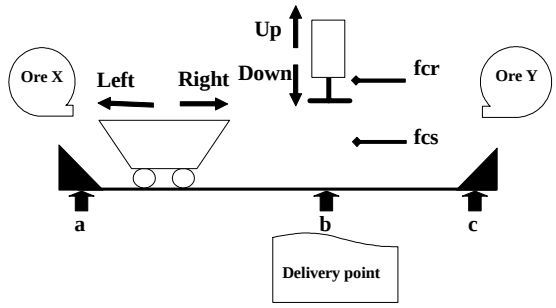

Figure 3: Exemple of application

the number of observable events and *m* is the number of normal states. Then, we determine the set of events Δ_k produced by the same sensor *k*. Let suppose that for the sensor 1: $\Delta_1 = \{\alpha_1, \alpha_4\}$and for the sensor 2: $\Delta_2 = \{\alpha_2, \alpha_3\}$. For each event, we put 0 for all the states sources of this event as well as the events produced by the same sensor. We put 1 for the other colons. This matrix will be used to know if a system is detectable or not and then if it is diagnosable or not. Indeed, the event, α_2, indicates that the state 1 was normal, but we need to wait the event α_4 to know if the state 2 was normal. For the example of the figure 2, we can find:

$$M = \begin{bmatrix} \Sigma_o / S_N & 1 & 2 & 3 & 4 \\ \alpha_1 & 0 & 1 & 1 & 0 \\ \alpha_2 & 1 & 0 & 0 & 1 \\ \alpha_3 & 1 & 0 & 0 & 1 \\ \alpha_4 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (3)$$

The application of (1) on (3) we can find that the system is detectable iff:

$$And(OR_j(M_i : i = 1..n) : j = 1..m) = 1 \quad (4)$$

If the system is detectable, the application of (3) on (2) determines if the system is diagnosable :

$$And(\overline{XOR_i}(M_j : j = 1..m) : i = 1..n) = 1 \quad (5)$$

We can find that the system of the figure 2 is diagnosable. To calculate the delay to realize the diagnosis, we use the following equation:

$$Delay = Max(NO1_i(M_j \quad j = 1..m) : i = 1..n) \quad (6)$$

where $NO1_i$ is the number of ones in each line *i*. As an example, to detect and localize a failure in the state 2, failure f_4, we need to wait that the event α_4 takes place. Thus the delay is equal to 2 observable events. In applying the same manner for the other lines, we can find that the system is 2-diagnosable for all the failures of the figure 2.

6 APPLICATION OF THE PROPOSED METHOD

We will apply the notion of diagnosabilty to the following example of a wagon with an electric actuator with two senses of movements: left and right, and a double effect cylinder: up and down. Three sensors *a*, *b* and *c* are used to determine the wagon position and two sensors *fcr* and *fcs* to determine the cylinder position as it is illustrated in figure 3. The schedule conditions are well defined and the following hypotheses are verified:

- There is one product (i.e. one wagon), each actuator has its proper sensors and each sensor is used by one actuator. The relax of this hypothesis will be study in other paper,
- An accepted response time is defined for each actuator as well as for the process by the designer,
- The wagon inertia is null.

The figure 4 shows the functional models of the actuator (wagon) and the double effect cylinder according to the SFC (Philippot, 2005).

The diagnosability matrix for the actuator, M_{act}, and the cylinder, M_{cyl}, for the sensor failures are:

$$M_{act} = \begin{bmatrix} \Sigma_o / S_N & 1 & 4 & 7 & 11 & 19 & 8 & 12 & 16 \\ \downarrow a & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ \uparrow b & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ \downarrow b & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ \uparrow c & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ \downarrow c & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ \uparrow a & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$M_{cyl} = \begin{bmatrix} \Sigma_o / S_N & 1 & 4 & 11 & 8 \\ \downarrow f_{cr} & 0 & 1 & 1 & 0 \\ \uparrow f_{cs} & 1 & 0 & 0 & 1 \\ \downarrow f_{cs} & 1 & 0 & 0 & 1 \\ \uparrow f_{cr} & 0 & 1 & 1 & 0 \end{bmatrix}$$

We can find from M_{act}, and M_{cyl}, that the cylinder is

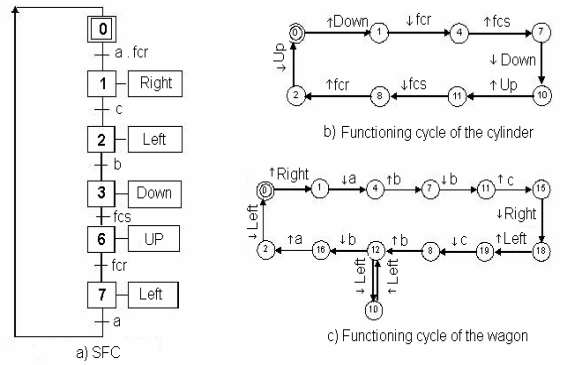

Figure 4: Functional models of the wagon and the cylinder for the application example

diagnosable and it is 2-diagnosable system and the actuator (wagon) is diagnosable and it is 6-diagnosable system. As an example, while being in the state 11, figure 4, a failure has occurred in the sensor *a*. We need to wait the occurrence of the event *b* of the state 4 to detect and isolate this failure.

7 CONCLUSION

In this paper, an algorithm to determine if a Discrete Event System (DES) is diagnosable or not, for a set of failures and according to a set of observable events, is presented. This algorithm treats the case of failures modelled by non-observable events for both actuators and sensors. The failures modelled by observable events can be also treated by this algorithm and the detection and isolation will be realized without any delay. This algorithm uses the notion of events to determine if a permanent failure has occurred. At the same time and to find a remedy to the problem of initialization of the system and the diagnoser, it uses the notion of state, to determine if a failure has occurred before the initialization of the diagnoser. This diagnosis is realized within a bounded delay in basing on the sensors outputs and the events sequences and their occurrence times. This algorithm was tested successfully on an example of manufacturing system. Firstly, this algorithm has shown that the system is diagnosable for the set of sensors and actuators failures and according to the set of observable events. Then a method to diagnose DES has applied, it has diagnosed several simulated failures within a bounded delay maximally equal to 6 events.

REFERENCES

Cassandras, C. G., Lafortune, S., 1999. *Introduction to Discrete Event Systems*, Kluwer Academic Publisher.

Deepa Pandalai, N., Holloway, L. E., 2000. Template Languages for Fault Monitoring of Timed Discrete Event Processes, In *IEEE Transactions On Automatic Control 45(5).*

Garcia, H. E., Yoo, T. S., 2005. Model-based detection of routing events in discrete flow networks, *Automatica 41*.

Holloway, L. E., Chand, S., 1994. Time templates for discrete event fault monitoring in manufacturing systems, In *American Control Conf.*, Baltimore, MD.

Lin, F., 1994. Diagnosability of Discrete Event Systems and its Applications, In *Discrete Event Dynamic Systems4*, Kluwer Academic Publishers, Boston, USA.

Perrow, C., 1984. *Normal Accidents:Living with High Risk Technologies*, Basic Books, Inc., New York.

Philippot, A., Sayed Mouchaweh, M., Carré-Ménétrier, V., 2005. Multi-models approach for the diagnosis of Discrete Events Systems, In *IMACS'05, International conference on Modelling, Analyse and Control of Dynamic Systems*, Paris-France.

Philippot, A., Tajer, A., Gellot, F., Carre-Ménétrier, V., 2004. *Méthodologie de modélisation dans le cadre de la synthèse formelle des SED.* Conférence Internationale Francophone d'Automatique, Douz, Tunisie.

Ramadge, P., Wonham, W., 1987. Supervisory control of a class of discrete event processes, In *SIAM J. Control Optim.* 25(1).

Sampath, M., Sungupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D., 1994. Diagnosability of discrete event systems, In *11th Int. Conf. Analysis Optimization of Systems: Discrete Event Systems*, Sophia-Antipolis, France.

Su, R., 2004. *Distributed Diagnosis for Discrete-Event System*, Thesis of PhD, University of Toronto, Canada.

Tripakis S., 2001; Fault Diagnosis for Timed Automata, VERIMAG (www-verimag.imag.fr).

Wang, Y., 2000. *Supervisory Control of Boolean Discrete-Event Systems*, Thesis of Master of Applied Sciences, University of Toronto, Canada.

Zad, S. H., Kwong, R. H., Wonham, W. M., 2003. Fault Diagnosis in Discrete Event Systems: Framework and model reduction, *IEEE Transactions On Automatic Control 48(7).*

Zad, S. H., Kwong, R. H., Wonham, W. M., 1998. Fault diagnosis in discrete-event systems. In *CDC'98, IEEE Conference on Decision and Control*, Tampa, Florida, USA.