Comparing diagnosability in Continuous and Discrete-Event Systems

Marie-Odile Cordier
INRIA
Rennes, France

Louise Travé-Massuyès and **Xavier Pucel**
LAAS-CNRS
Toulouse, France

Abstract

This paper is concerned with diagnosability analysis, which proves a requisite for several tasks during the system's life cycle. The Model-Based Diagnosis (MBD) community has developed specific approaches for Continuous Systems (CS) and for Discrete Event Systems (DES) in two distinct and parallel tracks. In this paper, the correspondences between the concepts used in CS and DES approaches are clarified and it is shown that the diagnosability problem can be brought back to the same formulation using the concept of signatures. These results bridges CS and DES diagnosability and open perspectives for hybrid model based diagnosis.

1 Introduction

Diagnosis is an increasingly active research domain, which can be approached from different perspectives according to the type of system at hand and the required abstraction level. Although some recent works have considered diagnosis based on hybrid models [Williams and Nayak, 1996; Bénazéra *et al.*, 2002; Bénazéra and Travé-Massuyès, 2003], the Model-Based Diagnosis (MBD) community has developed specific approaches for Continuous Systems (CS) and for Discrete Event Systems (DES) in two distinct and parallel tracks. Algorithms for monitoring, diagnosis and diagnosability analysis have been proposed [Sampath *et al.*, 1995; Jiang *et al.*, 2001; Yoo and Lafortune, 2002; Cimatti *et al.*, 2003; Rozé and Cordier, 2002; Jeron *et al.*, 2006]. The formalisms and tools are quite different : the CS community makes use of algebro-differential equation models or qualitative abstractions whereas the DES community uses finite-state formalisms. For diagnosability analysis, the CS approaches generally adopt a *state-based diagnosis* point of view in the sense that diagnosis is performed on a snapshot of observables, i.e. one observation at a given time point. The DES approaches perform *event-based diagnosis* and achieves *state tracking*, which means dynamic diagnosis reasoning achieved across time.

This paper is concerned with diagnosability analysis, which proves a requisite for several tasks during the system's life cycle, in particular instrumentation design, end-of-line testing, testing for diagnosis, etc. In spite of quite different frameworks, it is shown that the diagnosability assessment problem stated on both sides can be brought back to the same formulation and that common concepts can be proposed for proving diagnosability definitions equivalent.

2 DES and CS modelling approaches

This section presents the different theories used to model DESs and CSs. The principles underlying DES and CS model based diagnosis are given and diagnosability is introduced on both sides. Both approaches rely on the analysis of the observable consequences of faults, i.e. symptoms.

The main difference between DES and CS diagnosability analysis processes is that the order of appearance of the symptoms is only taken into account in the DES approach. In the CS approach, fault occurrence assumes immediate and simultaneous observation of the symptoms, while in the DES approach diagnosis relies on the observation of a sequence of symptoms after fault occurrence. Proof is given that, assuming the system observed a sufficiently long time, diagnosability conditions for DES and CS are conceptually equivalent.

2.1 The models

DES model

A DES is modelled by a language $L_{sys} \subseteq E^*$ where E is the set of system events. $L_{sys}$is prefix-closed, and can be described by a regular expression, or generated by a finite state automaton $G = (Q, E, T, q_0)$ where Q is the set of states, E the set of events, $T \subseteq (Q \times E \times Q)$ the transition relation and q_0 the initial state. Each trajectory in the automaton corresponds to one word of the language, and represents a sequence of events that may occur in the system. The set of events E is partitioned into observable and unobservable events : $E = E_o \cup E_{uo}$, and a set of faults $E_f \subseteq E_{uo}$ is given. The diagnosis process aims at detecting and assessing the occurrence of unobservable fault events from a sequence of observed events. The set OBS is defined as the set of all the possible observable events sequences, i.e, $OBS = \{(e_1 e_2 \dots e_n)\}$ where n is any positive integer.

In this article, it is assumed that the automaton is deterministic ($T : Q \rightarrow E \times Q$ is a function), generates a live language (every state has at least one outgoing transition), and contains no cycle of unobservable events.

The diagnosis process makes use of a projection operation that removes all unobservable events from a trajectory. The inverse operation is applied to a set of observable events sequences and leads to the diagnoses. A fault is diagnosable when its occurrence is always followed by a bounded observable event sequence that cannot be generated in its absence (see definition 1).

CS model

The behavior model of a CS $\Sigma = (R, V)$ is generally described by a set of n relations R, which relate a set of m variables V. In a component-oriented model, these relations are associated to the system physical components, including the sensors. The set R is partitioned into behavioral relations which correspond to the internal components and observation relations which correspond to the sensors. The set of variables V is also partitioned into the set of observed variables O, whose corresponding value tuples are called *observations*, and the set of unobserved variables noted X.

Observation values, possibly processed into fault indicators, provide a means to characterize the system at a given time. In a pure consistency-based approach, in which only the normal behavior of the system is modelled, the designer may use the model to establish a set of Analytical Redundant Relations, which can be expressed as a set of residuals. In that case, the observations result in a boolean fault indicator tuple. In the following, we will refer without loss of generality to the observation tuples and define the set OBS as the set of all the possible observation tuples, i.e., $OBS = \{(o_1, o_2, ..., o_k)\}$ where k is the number of sensors. The observation value pattern is referred to as the *observed signature* whereas the expected value patterns for a given fault, obtained from the behavioral model, provide the *fault signature*. Note that several value patterns may correspond to the same fault, for example when the system undergoes several operating modes. The fault signature is hence defined as the set of all possible observable variable value tuples under the fault. The diagnosis process relies on comparing the observed signature with fault signatures. Fault signatures also allow one to test fault detectability.

2.2 The set of observables

In the case of DES, observations consist in a sequence of observable events, while in the case of CS, observations consist in a set of values for observable variables, with no ordering.

This paper focuses on comparing the notions based on observations that lead to diagnosability, making abstraction of the nature of the observations. It is shown that the concept of signatures can be defined in a way allowing to prove the equivalence of definitions. However, it does not imply that any system being diagnosable when modelled as a DES is diagnosable as a CS, due to the difference in the observations nature. The set of observables OBS is defined as the set containing all the observations that are possible for the system. It may represent the observations obtained from a DES (a set of ordered observable events) as well as those from a CS (a set of observable values).

3 Faults, diagnoses and fault signatures

This section contains formal definitions of faults, diagnoses, and fault signatures. The definitions of diagnosability rely on these (see next section).

3.1 Faults and diagnoses

The set of faults F_{sys} associated to a system is partitioned into n types of faults, the partition is noted F. The following properties hold :

- $\forall F_i, F_j \in F, F_i \cap F_j \neq \emptyset \Rightarrow i = j$
- $\bigcup_{i=0}^{n} F_i = F_{sys}$

The occurrence of one or several faults of one type is called a single fault. When faults of several types have occurred, the system is said to be under a multiple fault. The set of possible faults that may occur in a system is the power set of F, noted $\mathcal{P}(F)$. For example, $\emptyset$ describes the absence of faults, $\{F_i\}$ a single fault, and $\{F_i, F_j\}$ a multiple fault. All three examples are elements of $\mathcal{P}(F)$. Faults are assumed to be permanent.

A diagnosis consists in a set of fault candidates. When a diagnosis contains only one fault, it is said to be determinate, while if it contains several faults it is indeterminate. The set of all possible diagnoses is the power set of the set of faults, noted $\mathcal{P}(\mathcal{P}(F))$. For example, $\{\emptyset\}$, $\big\{\{F_i\}\big\}$ and $\big\{\{F_i, F_j\}\big\}$ are determinate diagnoses, while $\big\{\emptyset, \{F_i\}, \{F_j, F_k\}\big\}$ is an indeterminate diagnosis indicating that one of the three diagnosis candidates $\emptyset$, $\{F_i\}$ and $\{F_j, F_k\}$ have occurred.

3.2 Fault signatures

Establishing fault signatures is the main part of our diagnosability analysis process. This concept is commonly used in the CS approach, but less in DES. The CSs' notion of fault signature is generalized and extended to DESs, allowing one to write diagnosability criterions in a unified way.

In a general way, one can consider a fault signature as a function Sig associating a set of observables to each fault. $Sig : \mathcal{P}(F) \rightarrow \mathcal{P}(OBS)$

Continuous systems

The fault signature is a classical concept in the CS approach usually defined as follows. For a fault f of $\mathcal{P}(F)$, let OBS_f be the set of all possible tuples consisting of observed variable values under the fault f, regardless of time[1]. Then :

$$Sig(f) = OBS_f \in \mathcal{P}(OBS)$$

Discrete event systems

Fault signatures are based upon the projection over observable events, which are defined in a first step. They correspond to what is usually known as observable trajectories in the DES community.

Language projection The language projection over the set of observable events E_o, noted P_{obs}, to a language L, associates the language formed by the words of L restricted to the letters that are elements of E_o. For example if $L = \{e_1,\, e_1e_3,\, e_1e_2,\, e_2e_3,\, e_1e_2e_3\}$ and $E_o = \{e_1, e_2\}$, then

[1] Note that "under the fault f" means that exactly all the faults in f occured, and no faults out of f occured.

$P_{obs}(L) = \{e_1,\ e_1e_2,\ e_2\}$. The inverse projection P_{obs}^{-1}, defined on $\mathcal{P}(OBS)$, to a set of observable events sequences, associates the set of trajectories (which is a language) whose projections belong to the antecedent set :

$$\forall O \in \mathcal{P}(OBS),$$
$$P_{obs}^{-1}(O) = \left\{s \in L_{sys}, P_{obs}(\{s\}) \cap O \neq \emptyset\right\}$$

Fault language For each fault $f \in \mathcal{P}(F)$, the *f-language*, or L_f, describes all possible trajectories in which f occurs. L_f is defined as the subset of the system's automaton's language L_{sys}, restricted to the words containing at least one occurrence of every single fault event composing f, and no occurrence of any other fault event. L_f describes all possible scenarios in which f occurs. The words of the f-language are called *f-trajectories*.

Fault signature Because of our particular interest for diagnosability, among the set of f-trajectories, we pay special attention to those that can be obtained when the observation temporal window can be arbitrarily extended. This is done by considering, in L_f, only words that end in an infinite cycle. They are defined as the *maximal words*, and form the *maximal f-language L_f^{max}* of the fault. Formally, a trajectory s of L_f belongs to L_f^{max} if and only if $\exists t, u \in E^*, s = tu^\infty$. Notation u^∞ refers to the word built as an infinite concatenation of word u, i.e., every $u^n \in u^*$ is a prefix of u^∞.

For each fault $f \in \mathcal{P}(F)$, the projection of the maximal f-language L_f^{max} over the set of observable events is called the *f-signature*. Any f-signature is a subset of OBS as it is solely composed of observable events. With the above definitions, it is possible to define the signature function *Sig* as the function associating its f-signature to any fault $f \in \mathcal{P}(F)$:

$$\forall f \in \mathcal{P}(F), \quad Sig(f) = f\text{-signature} \in \mathcal{P}(OBS)$$

4 Diagnosability

Formal definitions of diagnosability according to the DES and CS approaches are now given.

4.1 Discrete Event Systems

We rely here on the (strong)[2]diagnosability definition as defined by [Sampath *et al.*, 1995].

DES (strong) Diagnosability : a DES is (strongly) diagnosable if and only if[3] :

$$\begin{aligned} \forall F_i \in F, \exists n_i \in \mathbb{N}, \forall s \in L_{sys}/(F_i \in s), \\ \forall t \in E^*/(st \in L_{sys}), \\ \|t\| \geq n_i \Rightarrow \forall u \in P_{obs}^{-1}\big(P_{obs}(st)\big), F_i \in u \end{aligned} \tag{1}$$

One can notice that the definitions are stated with respect to elements of F. The system is required to be diagnosable for each fault type, independently of the fact that they are single or multiple faults.

4.2 Continuous systems

In the CS approach, the classical definition of diagnosability is already given in terms of the fault signature concept as follows [Travé-Massuyès *et al.*, 2004].

CS (Strong) Diagnosability : a CS is (strongly) diagnosable if and only if :

$$\forall f_1, f_2 \in \mathcal{P}(F), f_1 \neq f_2, Sig(f_1) \cap Sig(f_2) = \emptyset \tag{2}$$

This definition applies to single or multiple faults and differs from the DES definitions in this respect. It is shown in the next section that this difference is not relevant and that the fault signature concept is a unifying concept allowing one to formally compare the two approaches.

5 Formal Comparison

In this section, we give the proof of equivalence between the diagnosability definition in the DES and CS approaches. We first prove that the DES definition can be extended to multiple faults, which provides a better insight into the definition interpretation.

As noted before, definition (1) is stated for elements of F, which corresponds to consider single faults. Let us extend it to multiple faults. The occurrence of a multiple fault f in a trajectory s is noted $\forall F_i \in f, s \cap F_i \neq \emptyset$. The diagnosability condition (1) is verified for each $F_i \in f$ with possibly different n_i values. Taking the largest value of all these n_i values as n_f, it can be easily shown that definition (1) is equivalent to definition (1′), which accounts explicitely for multiple faults $f = \{F_i\}$.

$$\begin{aligned} \forall f \in \mathcal{P}(F), \exists n_f \in \mathbb{N}, \\ \forall s \in L_{sys}/\big(\forall F_i \in f, F_i \in s\big), \\ \forall t \in E^*/(st \in L_{sys}), \|t\| \geq n_f \Rightarrow \\ \forall u \in P_{obs}^{-1}\big(P_{obs}(st)\big), \forall F_i \in f, F_i \in u \quad \blacksquare \end{aligned} \tag{1'}$$

This result shows that the DES diagnosability definition can be given in terms of faults (instead of fault types), whether single or multiple, like the CS diagnosability definition.

The equivalence between diagnosibility definitions is now proved by considering the assessment upon absence of faults in a diagnosable discrete events system.

Let us consider a diagnosable system, thus verifying (1), and trajectories of arbitrary length, in particular maximal trajectories which correspond to maximal words as defined in section 3.2. Let us consider such a maximal trajectory s belonging to the f-language L_f. It means that s contains at least one occurrence of every single fault event composing f and *no occurrence of any other fault*. s belongs thus to L_f^{max} and its projection over the set of observable events belongs to the f-signature. Now suppose that there exists a (maximal) trajectory u such that $P_{obs}(\{u\})$ equals $P_{obs}(\{s\})$ and that u contains at least one occurrence of a fault F_j which does not belong to f. By (1), it implies that all trajectories sharing the observable projection of u contain F_j, which is contradictory with our hypothesis about s. Thus, there does not exist any trajectory having the same observable projection as s and containing a fault not belonging to f. This proves that $\forall f_1, f_2 \in \mathcal{P}(F), f_1 \neq f_2, Sig(f1) \cap Sig(f2) = \emptyset$ which is exactly the definition (2) given in 4.2 for the Continous Systems. ■

[2] A definition for weak diagnosability is given in [Rozé and Cordier, 2002] for DES and in [Travé-Massuyès *et al.*, 2004] for CS

[3] The notation $F_i \in s$ means that s contains at least one fault event of F_i.

6 Operational comparison

This section contains an example that illustrates the concepts introduced before and compare the DES and CS diagnosability results.

6.1 Example

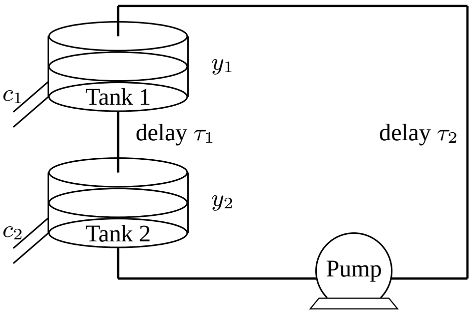


Figure 1: A water flow system

The system represented in figure 1 is inspired of [Puig *et al.*, 2005]. It is composed of two water tanks with heights y_1 and y_2, and a pump connected by a water flow channel. Both tanks supply consumers c_1 and c_2. The delays τ_1, respectively τ_2, correspond to the time needed for the water to reach tank2 from tank1, respectively tank1 from the pump. It has two operating modes : *pump on* and *pump off*. We consider faults in sensors y_1, y_2, c_1 and c_2, named respectively F_{y1}, F_{y2}, F_{c1} and F_{c2}.

To simplify, the example is limited to single faults and assume the system does not switch its operating mode between the occurrence of a fault and the appearance of its symptoms.

6.2 Continuous model, state-based diagnosis

The discretized and linearized non-linear dynamic equations are described in [Puig *et al.*, 2005]. From these equations, it is possible to derive two analytical redundancy relations of the form :

$r_1(t+\Delta t) = f_1\big(y_1(t), y_1(t+\Delta t), c_1(t), y_2(t-\tau_2)\big)$ $r_2(t+\Delta t) = f_2\big(y_2(t), y_2(t+\Delta t), c_2(t), y_1(t-\tau_1)\big)$

From these relations and considering when pump is off, r_1 is independent of y_2, the fault signature matrices shown in Figure 2 are deduced.

	F_{y1}	F_{y2}	F_{c1}	F_{c2}
r_1	1	1	1	0
r_2	1	1	0	1

Pump on mode

	F_{y1}	F_{y2}	F_{c1}	F_{c2}
r_1	1	0	1	0
r_2	1	1	0	1

Pump off mode

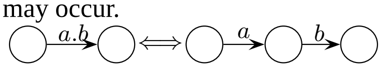
Figure 2: Fault signature matrices for the system

From the fault signature matrices, the system is not diagnosable (for example, the observable $(p_{on}, r_1 = 1, r_2 = 1)$ belongs to two fault signatures).

6.3 Discrete event model, dynamic diagnosis

For the DES model of the system, the following events are used : p_{on},p_{off}, fired when the pump is turned on or off ; F_S fired when a fault occurs on sensor S ; r_1, r_2 fired when analytical redundancy relations r_1 and r_2, are violated.

The automaton is shown in figure 3. An arc labelled $a.b$ represents two arcs labelled a and b, a leading to a state in which only b may occur.



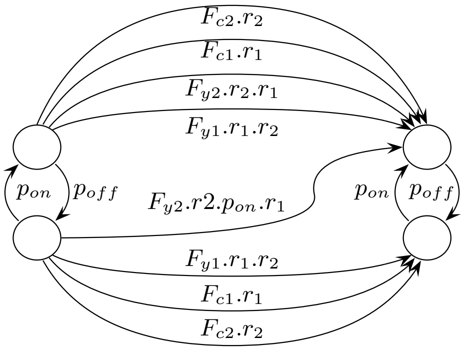


Figure 3: Automaton describing the system

Fault	Signature
$\emptyset$	$(p_{on}.p_{off})^\infty$
F_{c1}	$(p_{on}.p_{off})^*.\mathbf{r_1}.(p_{on}.p_{off})^\infty$ $(p_{on}.p_{off})^*.\mathbf{p_{on}}.\mathbf{r_1}.(p_{off}.p_{on})^\infty$
F_{c2}	$(p_{on}.p_{off})^*.\mathbf{r_2}.(p_{on}.p_{off})^\infty$ $(p_{on}.p_{off})^*.\mathbf{p_{on}}.\mathbf{r_2}.(p_{off}.p_{on})^\infty$
F_{y1}	$(p_{on}.p_{off})^*.\mathbf{r_1}.\mathbf{r_2}.(p_{on}.p_{off})^\infty$ $(p_{on}.p_{off})^*.\mathbf{p_{on}}.\mathbf{r_1}.\mathbf{r_2}.(p_{off}.p_{on})^\infty$
F_{y2}	$(p_{on}.p_{off})^*.\mathbf{r_2}.\mathbf{p_{on}}.\mathbf{r_1}.(p_{off}.p_{on})^\infty$ $(p_{on}.p_{off})^*.\mathbf{p_{on}}.\mathbf{r_2}.\mathbf{r_1}.(p_{off}.p_{on})^\infty$

Figure 4: Fault signatures (discriminant subwords are bolded).

From the automaton and following section 3.2, it is possible to build the signatures for all the faults (see Figure 4). Recall that all the events except faults are observable. The fault signatures are disjoint sets, the system is hence diagnosable.

6.4 Results

This example shows that, although DES and CS diagnosability definitions are formally equivalent, operationnal diagnosability assesment critically depends on the nature of observables.

In the CS approach, diagnosability is not achieved, as fault signatures are not disjoint. $(p_{on}, r_1 = 1, r_2 = 1)$ is a signature for both F_{y1} and F_{y2}, and $(p_{off}, r_1 = 0, r_2 = 1)$ is a signature for both F_{y2} and F_{c2}.

In the DES model, in the *pump on* mode, the symptoms $r_1 = 1$ and $r_2 = 1$ appear in the order $(r_1 r_2)$ for F_{y1} and in reverse order $(r_2 r_1)$ for F_{y2}. Taking this order into account permits fault discrimination between F_{y1} and F_{y2} in dynamic diagnosis. In addition, in the *pump off* mode, both F_{y2} and

F_{c2} are followed by the r_2 symptom, but only in the case of F_{y2}, a p_{on} command will be followed by the r_1 symptom. Notice that diagnosability stands on the assumption that the pump will be turned on some time : it is only after the p_{on} command that the faults can be discriminated.

7 Related work

In the context of continuous systems, diagnosability analysis is stated in terms of detectability and isolability [Chen and Patton, 1994]. [Basseville, 2001] reviews several definitions of fault detectability and isolability and distinguishes two types of definitions, namely intrinsic definitions that do not make any reference to a particular residual generator and performance-based definitions. In [Staroswiecki and Comtet-Varga, 1999], the conditions for sensor, actuator and component fault detectability are given for algebraic dynamic systems and isolability is discussed. Diagnosability analysis for continous systems is often focussed on finding the optimal sensor placement as [Travé-Massuyès *et al.*, 2001], proposing a structural approach, or [Yan, 2004], and [Tanaka, 1989]. [Frisk *et al.*, 1993] also follow a structural approach and show how different levels of knowledge about the faults may influence the fault isolability properties of the system. In [Travé-Massuyès *et al.*, 2004], a definition for diagnosability in terms of fault signatures is proposed and is the one used in this paper.

In the DES context, the first definitions have been proposed in [Sampath *et al.*, 1995]. Checking diagnosability is computationally complex and polynomial time algorithms have been designed to cope with this problem [Jiang *et al.*, 2001; Yoo and Lafortune, 2002]. In [Cimatti *et al.*, 2003], formal verification of diagnosability is based on model-checking techniques. More recently, [Jeron *et al.*, 2006] propose a generalization of diagnosability properties to supervision patterns (describing various patterns involving fault events).

To our knowledge, there is no existing work comparing and/or unifying diagnosability approaches coming from the CS and DES communities. Some diagnosis algorithms have been proposed for hybrid systems but diagnosability conditions have not been exhibited for such systems and this is one of our goals for future work. This paper is a direct continuation of the work done with the Imalaia group and devoted to bridge the gap between the two communities [Cordier *et al.*, 2004] by comparing their respective approaches to model-based diagnosis.

8 Conclusion

In this paper, we propose a formal framework to compare in an adequate way the diagnosability definitions from the CS and DES community. The signature concept is generalized to trajectories and allows to prove equivalence of the diagnosability definitions. The main point is the way observations are defined, in a static way in the CS approach and as partially ordered sets (sequences) in the DES approach. In one hand, when temporal information is necessary to discriminate faults, the DES approach gives better results. In another hand, it requires to wait a certain amount of time, before having the result. In practical applications, this delay has to be estimated and must be realistic wrt existing risks and decisions to be taken. Another view is to enrich CS signatures with temporal information [Puig *et al.*, 2005].

Having a common diagnosability analysis approach for both state-based and dynamic diagnosis also opens perspective for analysing hybrid systems diagnosability.

References

[Basseville, 2001] M. Basseville. On fault detectability and isolability. *European Journal of Control*, 7(8):625–637, 2001.

[Bénazéra and Travé-Massuyès, 2003] E. Bénazéra and L. Travé-Massuyès. The consistency approach to the on-line prediction of hybrid system configurations. *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'03), Saint-Malo (France)*, 2003.

[Bénazéra *et al.*, 2002] E. Bénazéra, L. Travé-Massuyès, and P. Dague. State tracking of uncertain hybrid concurrent systems. *Proceedings of the International Workshop on Principles of Diagnosis(DX'02)*, pages 106–114, 2002.

[Chen and Patton, 1994] J. Chen and R.J. Patton. A reexamination of fault detectability and isolability in linear dynamic systems. *Proceedings of the 2nd Safeprocess Symposium, Helsinki (Finland)*, 1994.

[Cimatti *et al.*, 2003] A. Cimatti, C. Pecheur, and R. Cavada. Formal verification of diagnosability via symbolic model checking. *Proceedings of IJCAI'03*, pages 363–369, 2003.

[Cordier *et al.*, 2004] M.-O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus analytical redundancy relations : A comparative analysis of the model-based diagnostic approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetics - Part B.*, 34(5):2163–2177, 2004.

[Frisk *et al.*, 1993] E. Frisk, D. Dstegr, M. Krysander, and V. Cocquempot. Improving fault isolability properties by structural analysis of faulty behavior models: application to the damadics benchmark problem. *Proceedings of the 1st SafeProcess Symposium*, 1993.

[Jeron *et al.*, 2006] T. Jeron, H. Marchand, and M-O. Cordier. Motifs de surveillance pour le diagnostic de systèmes évènements discrets. In *Proceedings of RFIA'2006*, 2006.

[Jiang *et al.*, 2001] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[Puig *et al.*, 2005] V. Puig, J. Quevedo, T. Escobet, and B. Pulido. On the integration of fault detection and isolation in model based fault diagnosis. In *Proceedings of DX'05*, pages 227–232, 2005.

[Rozé and Cordier, 2002] L. Rozé and M.-O. Cordier. Diagnosing discrete-event systems : extending the "diagnoser approach" to deal with telecommunication networks. *Journal on Discrete-Event Dynamic Systems : Theory and Applications (JDEDS)*, 12(1):43–81, 2002.

[Sampath *et al.*, 1995] M. Sampath, R. Sengputa, S. Lafortune, K. Sinnamohideen, and D. Teneketsis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40:1555–1575, 1995.

[Staroswiecki and Comtet-Varga, 1999] M. Staroswiecki and G. Comtet-Varga. Fault detectability and isolability in algebraic dynamic systems. *Proceedings of the European Control Conference*, 1999.

[Tanaka, 1989] S. Tanaka. Diagnosability of systems and optimal sensor location, 1989.

[Travé-Massuyès *et al.*, 2001] L. Travé-Massuyès, T. Escobet, and Rob Milne. Model-based diagnosability and sensor placement application to a frame 6 gas turbine subsystem. *Proceedings of IJCAI'01*, pages 551–556, 2001.

[Travé-Massuyès *et al.*, 2004] L. Travé-Massuyès, T. Escobet, and X. Olive. Model-based diagnosability. *Internal Report LAAS N04080, Janvier 2004, 12p. to appear in IEEE Transactions on System, Man and Cybernetics, Part A*, 2004.

[Williams and Nayak, 1996] B. C. Williams and P. P. Nayak. A model-based approach to reactive self-configuring systems. *Proceedings of AAAI-96, Portland, Oregon*, pages 971–978, 1996.

[Yan, 2004] Y. Yan. Sensor placement and diagnosability analysis at design stage. *Notes of the MONET Workshop on Model-Based Systems, ECAI'04*, 2004.

[Yoo and Lafortune, 2002] T. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Trans. on Automatic Control*, 47(9):1491–1495, 2002.