CHARACTERIZATION OF REQUIREMENTS AND COSTS FOR THE DIAGNOSABILITY OF DISTRIBUTED DISCRETE EVENT SYSTEMS

Pauline Ribot * **Yannick Pencolé** *
Michel Combacau *

* *CNRS-LAAS, University of Toulouse*

Abstract: We address the problem of fault diagnosabilty in a distributed discrete-event systems. Previous works mainly propose different ways to check whether a fault is diagnosable or not. Nowadays, due to the complexity of the engineered systems, this checking is not enough and a better feedback is required in order to redesign and guarantee the diagnosability of a fault. This paper defines the problem of the automatic computation of design requirements for the diagnosability of distributed discrete event system as a cost optimization problem.

Keywords: Diagnosability, Design Requirements, Distributed Discrete-Event Systems

1. INTRODUCTION

We address the problem of fault diagnosis in a distributed discrete-event systems. This problem has been studied for many years, Sampath et al. (1995), Debouk et al. (2002), Lamperti and Zanella (2003), Console et al. (2002), Pencolé and Cordier (2005). In these previous works, the objective is to model the system and to apply monitoring algorithms on it but they all share the same weakness: none of them takes into account any diagnosablility issue about the system. If a diagnosability analysis is performed on the system, then the diagnosis algorithm is more efficient and less costly because new information is taken into account before implementing it.

In this paper, we adopt the following point of view. Due to the complexity of new large systems and the new requirements such as maintenance, reliability or security, the diagnosability study must be performed at the design stage of the system. One difficulty in the design of a distributed system is that it is usually conceived by different designers which are in charge of only a part of the system. It follows that the integration of the different parts of the system is very complex and has a direct impact on the difficulty to guarantee the diagnosability objective of the whole system.

Our main aim is to determine the characteristics and the modifications which could be useful for designers to improve and to guarantee some diagnosability objectives. This analysis relies on both the design of the distributed system itself (design and integration of components) and the design of the monitoring architecture in charge of diagnosing the system. We propose to formalize this problem as a cost optimization problem in a distributed framework. We then show the relationship between optimizing the cost of the design of the distributed system and optimizing the choice for a suitable monitoring architecture.

The paper is organized as follows. Section 2 recalls formal background on fault diagnosis and diagnosability. Section 3 defines the optimization cost problem for guaranteeing diagnosability in a distributed discrete-event system. Section 4 introduces a methodology that determines, given a prespecified system, a set of design requirements for diagnosability that takes into account the inherent characteristics of a distributed system and minimizes the implementation cost.

2. BACKGROUND

2.1 Formal definition of the DES formalism

Our study is placed in a discrete event systems framework for model-based diagnosis as defined in Sampath et al. (1995). This framework has been developed for several years and is used for different application types. Moreover, large number of properties have already been demonstrated within this framework.

Let us consider a distributed system Γ composed of interacting components $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$ which evolve by the occurrence of events. This system can be modelled as a set of automata, each automaton representing the model of a component (i.e. a local model).

Definition 1. (Local model). A local model Γ_i is an automaton $\Gamma_i = (Q_i, \Sigma_i, T_i, q_{0_i})$ where:

- Q_i is a finite set of states;
- Σ_i is the set of events occurring on Γ_i;
- $T_i \subseteq Q_i \times \Sigma_i \times Q_i$ is the set of transitions;
- q_{0_i} is the initial state.

Different events can occur on a component. The set Σ_i is partitioned into two disjoint sets : Σ_{l_i}, the set of events local to the component Γ_i and Σ_{c_i}, the set of interactive events which allow the communication between the different components. Local events of Σ_{l_i} can be observable or unobservable : $\Sigma_{l_i} \subseteq \Sigma_{o_i} \cup \Sigma_{uo_i}$, where Σ_{o_i} (resp. Σ_{uo_i}) is the set of observable (resp. unobservable) events. Σ_{f_i} denotes the set of fault events occurring on the component Γ_i which are to be diagnosed. Our diagnosis algorithm is based on the location of fault events on components, so we suppose that $\Sigma_{f_i} \subseteq \Sigma_{l_i}$. In comparison with the framework defined in Sampath et al. (1995), we authorize some interactive events to be observable : $\Sigma_{c_i} \subseteq \Sigma_{o_i} \cup \Sigma_{uo_i}$.

A subsystem γ is a non-empty set $\{\Gamma_{i_1}, \ldots, \Gamma_{i_m}\}$ of m components of the system. The behavior of the subsystem γ is modelled by an automaton $\|\gamma\|$ obtained from the classical synchronization operation, denoted $\|$, that is the automata product synchronized on the interactive events of γ : $\|\gamma\| = \Gamma_{i_1}\| \ldots \|\Gamma_{i_m}$. In the following, we denote $\Sigma_\gamma = \bigcup_{j=1}^m \Sigma_{i_j}$, $\Sigma_{f_\gamma} = \bigcup_{j=1}^m \Sigma_{f_{i_j}}$, $\Sigma_{c_\gamma} = \bigcup_{j=1}^m \Sigma_{c_{i_j}}$, $\Sigma_{o_\gamma} = \bigcup_{j=1}^m \Sigma_{o_{i_j}}$ and $\Sigma_{l_\gamma} = \bigcup_{j=1}^m \Sigma_{l_{i_j}}$. Each behavior of $\|\gamma\|$ is a sequence of the language $L(\gamma) \subseteq \Sigma_\gamma^*$, where Σ_γ^* denotes the Kleene closure of the set Σ_γ. The behavior of the system is explicitly represented by $\|\Gamma\|$.

2.2 Definition of a diagnoser for a subsystem

Let F be a fault event occurring on a component Γ_i which belongs to a subsystem γ, the diagnoser of the subsystem γ in charge of diagnosing F is denoted Δ_γ. After any sequence of observations σ emitted by γ, the diagnoser Δ_γ is a function that provides at any time a diagnosis information $\Delta_\gamma(F, \sigma)$ which is one of the following three types.

- $\Delta_\gamma(F, \sigma) = F$-sure : in every behavior of the subsystem γ which explains the sequence of observations σ, the fault F has occurred.
- $\Delta_\gamma(F, \sigma) = F$-safe : none of behaviors of γ which explains the sequence of observations σ contains the fault F.
- $\Delta_\gamma(F, \sigma) = F$-ambiguous : some behaviors of γ which explain the sequence of observations σ contains the fault F, some do not.

Generally the diagnoser is defined from the whole system Γ like in Sampath et al. (1995).

2.3 Definition of diagnosability on a subsystem

Diagnosability defines a property that measures the ability of the monitoring system to diagnose faults occurring in the supervised system. We rephrase the diagnosability definition from Pencolé (2005).

Definition 2. A fault event F is locally diagnosable in a subsystem γ if any of its occurrence on a component of γ is always followed by a finite sequence of observations such that the diagnosis of Δ_γ is F-sure.

Several tools allow the checking of diagnosability like the global diagnoser in Sampath et al. (1995). Our definition of diagnosability is similar to the one introduced by Sampath et al. (1995) but is applied to any subsystem γ. The following property states the relationship between local and global diagnosability.

Property 1. Under the assumption of observation fairness [1], if a fault F is locally diagnosable on a subsystem then F is diagnosable.

By this property, it may be unnecessary to observe the whole system to diagnose a fault F occurring

[1] Observation fairness means that any observable component will always emit observations after a finite time.

on a component of Γ. Observing only a subsystem γ can be sufficient.

3. DESIGN FOR DIAGNOSABILITY

The objective is to establish requirements for designers to ensure the diagnosability of the different parts of the distributed system. Two different ways can be considered to reach the objective. The first one consists in providing requirements for the design of a diagnosable system (diagnosability analysis is performed at the design stage) and the second one relies on a feedback to design which increases the system diagnosability. Both issues are related to the problem of assistance to the design like in Pencolé (2005).

To ensure diagnosability of the system, some modification operations on the local model of subsystems have to be considered depending on the monitoring architecture. These operations consist in increasing the subsystem observability or changing the subsystem structure.

3.1 Types of modifications for the system design

A subsystem γ is made diagnosable by authorizing modifications on its model. The first operation consists in increasing the subsystem observability by selecting optimum types, location and number of sensors. This operation tends to be similar to a sensor selection problem like in Debouk et al. (1999) or Jiang et al. (2003). The second operation acts directly on the subsystem structure by adding a new transition for example.

The possible operations to make a subsystem γ diagnosable are enumerated hereafter. All these modifications have a physical signification.

(1) Observing an event e (that may be a fault) which occurs in the subsystem γ: $e \in \Sigma_{l_\gamma} \cap \Sigma_{o_\gamma}$. Making an event observable means to add a sensor that is able to detect the occurrence of such an event on a component in the studied subsystem.

(2) Observing an event e which occurs on an other component α in the neighborhood: $e \in \Sigma_{l_\alpha} \cap \Sigma_{o_\alpha}$, where $\alpha \notin \gamma$. To observe an event on a component in the neighborhood of the subsystem, we need to place a sensor on this component and to consider a communication protocol in the monitoring architecture.

(3) Observing an interactive event i between the subsystem γ and an other component α: $i \in \Sigma_{c_\gamma} \cap \Sigma_{c_\alpha} \cap \Sigma_{o_\gamma} \cap \Sigma_{o_\alpha}$, where $\alpha \notin \gamma$. To make an interactive event observable consists in putting a sensor on the communication bus between two components or on a middleware for example.

(4) Observing an event $e \in \Sigma_{l_\gamma}$ only if given conditions *cond* are verified by introducing two new events *cond*:$e \in \Sigma_{o\gamma}$ and $\neg$*cond*:$e \notin \Sigma_{o_\gamma}$. To observe an event in a given condition we need to have a specific sensor which can be controlled on-line depending on other information. This sensor allows an active acquisition of information like in Thorsley and Teneketzis (2004).

(5) Adding/Deleting a transition t in the model of γ. New sensors can be generated like an alarm sensor after a given sequence of observations for example or new protocols can be implemented (communication protocol for instance). Transitions are reorganized by this new implementation and some of them may be deleted.

3.2 Specification of the diagnosis algorithm

In order to have a diagnosable system, the monitoring system needs to have information from components of the system. Accessing to these information resources has for the monitoring system a cost C_M which is induced by the choice of the diagnosis architecture type. The goal is to make the system diagnosable by optimizing the cost of modifications listed above and the cost of the monitoring system.

There mainly exist three different types of diagnosis architecture. In a centralized diagnosis architecture like in Sampath et al. (1995), all information from the components are centralized in the same place. In this architecture type, the data analysis is complex because all information are collected by the monitoring system without any processing. A lot of memory resources and communications are necessary for the monitoring system and induce the cost C_M. A centralized architecture may be not well suited for distributed systems. For large distributed systems it is more natural to adopt a decentralized or a distributed diagnosis architecture. In a decentralized diagnosis architecture, diagnosis decisions are sent from local diagnosers, which do not necessary communicate with each other. The different diagnosis information are then merged by a coordinator in order to establish a global diagnosis of the whole system. In a distributed architecture, there is a diagnoser per component that performs its own diagnosis. In order to get globally consistent diagnoses, the diagnosers need to exchange a lot of information. This communication between the diagnosers induces a bandwidth cost which is included in the cost of the monitoring algorithm C_M.

3.3 Problem formalization like a cost optimization problem

To each modification on the system is associated a cost which is listed in a dictionary like in the Table 1. In this dictionary, observing an event is denoted by *Obs* and adding (resp. deleting) a transition in the model structure is denoted by *Add* (resp. *Del*). Some modifications on the system are infeasible. An infinite cost is associated to such modifications like for example the observation of some fault events. The goal is to give some requirements to the designers about how to make a system diagnosable by minimizing the total cost of operations on the system, denoted C_D, and minimizing the cost of the monitoring system algorithm C_M:

$$C_G = min \sum_{i=1}^{p} (C_{D_i} + C_{M_i}), \tag{1}$$

where p is the number of faults that may occur in the system and for which diagnosability must be guaranteed.

Operation		Cost
1	$Obs(e)$, $e \in \Sigma_{l_\gamma}$	$c_l(e)$
2	$Obs(e)$, $e \in \Sigma_{l_\alpha}$ where $\alpha \notin \gamma$	$c_e(e)$
3	$Obs(i)$, $i \in \Sigma_{c_\gamma}$	$c_i(i)$
4	$Obs(e, cond)$, $e \in \Sigma_{l_\gamma}$ and $e \notin \Sigma_{o_\gamma}$	$c_d(e)$
5	$Add(t)$, $Del(t)$	$c_t(t)$

Table 1. Dictionary of operations costs for a subsystem γ

The challenge is to determine a trade-off between both costs. For example, one possible modification is to observe an event on another component in the neighborhood. For this purpose, the monitoring system needs to use a communication protocol to get back the observation from the component. So we have to consider the cost C_M of the algorithmic procedure to communicate information to the monitoring system in addition to the cost directly associated to the sensor.

The second difficulty in the cost optimization problem is that all fault events $F \in \Sigma_f$ must be considered. Since the components of the system communicate by interactive events from Σ_c, it could be interesting to observe interactions even if this event type have a high cost in the dictionary because its observation could bring useful information to diagnose more than one fault event and thus reduces the monitoring cost. We have to compare the cost of observing an interaction with the cost of observing a certain number of events.

We could imagine that it is more complicated to modify the system structure but, in some situations, we would prefer to add a new transition rather than to observe a certain number of events or an event far from the component. It may be more economic to add new transition than to place a lot of sensors.

4. ACCURACY AND DIAGNOSABILITY

This section shows how accuracy property can help to establish a methodology for providing requirements and their costs to designers by indicating the part of the system to modify. In this section, we consider that a specification of the system already exists and the methodology provides a feedback as design requirements.

4.1 Accuracy

A subsystem γ is said to be accurate if it is sufficient to observe it in order to provide a diagnosis which is globally consistent. Accuracy is formally defined as follows. Let Δ_γ be the diagnoser of the subsystem γ and $\sigma_\gamma = P_{\Sigma_{o_\gamma}}(\sigma)$ be the projection of the sequence $\sigma \in \Sigma_o^*$ on the observable events of γ. The projection operation can be recursively defined as follows. Let ϵ denote the empty sequence in Σ^*.

Definition 3. (Projection). The projection operation $P_{\Sigma'} : \Sigma^* \to \Sigma^{'*}$ is such that $P_{\Sigma'}(\epsilon) = \epsilon$ and for all $uv \in \Sigma^*$, $u \in \Sigma$,

$$P_{\Sigma'}(uv) = \begin{cases} uP_{\Sigma'}(v) & \text{if } u \in \Sigma' \\ P_{\Sigma'}(v) & \text{otherwise.} \end{cases}$$

Definition 4. The subsystem γ is accurate for a fault event $F \in \Sigma_{f_\gamma}$ iff there exists a diagnoser $\triangle_\gamma$ such that

$$\forall \sigma \in \Sigma_o^*,\ \Delta(F, \sigma) = \Delta_\gamma(F, \sigma_\gamma). \tag{2}$$

With an accurate subsystem, the monitoring system does not require information from any other component. Independently from the diagnosability, finding an accurate subsystem is very interesting because it is a mean to bound the cost of the monitoring algorithm C_M. We can always find an accurate subsystem (the biggest one is the global system Γ itself).

4.2 Methodology

This section presents a methodology that locates a subsystem and proposes some modifications in order to make the subsystem diagnosable and accurate with minimal costs.

4.2.1. How to make a system accurate

A simple way to make a system accurate is to observe its interactions. Let Σ_{γ}^{int} be the set of events of γ that interact with the other components : $\Sigma_{\gamma}^{int} = \Sigma_{c_\gamma} \cap \Sigma_{c_{\Gamma\setminus\gamma}}$.

Property 2. Let γ be a subsystem, Δ_γ is F-accurate for all $F \in \Sigma_{f_\gamma}$ if $\Sigma_{\gamma}^{int} \subseteq \Sigma_o$.

Proof: Let F be a fault local to γ (i.e. $F \in \Sigma_{f_\gamma}$). To show the accuracy of Δ_γ, we firstly suppose that there exists a global and observable sequence σ such that the global diagnosis is $\Delta(F,\sigma) = F$-sure and the local diagnosis is $\Delta_\gamma(F,\sigma_\gamma) = F$-ambiguous. $\Delta(F,\sigma) = F$-sure therefore in every sequence seq of $\|\Gamma\|$ that explains σ (i.e. $P_{\Sigma_o}(seq) = \sigma$), F has occurred. $\Delta_\gamma(F,\sigma_\gamma) = F$-ambiguous so it means that there exists at least a local sequence seq' in $\|\gamma\|$ explaining σ_γ (i.e. $P_{\Sigma_{o_\gamma}}(seq') = \sigma_\gamma$) in which F has not occurred. Since $\sigma_\gamma = P_{\Sigma_{o_\gamma}}(\sigma)$ and $\Sigma_{\gamma}^{int} \subseteq \Sigma_o$, this local sequence seq' is necessarily globally consistent and, therefore, is part of at least one global sequence explaining σ, hence a contradiction. Let suppose now the existence of a sequence σ such that $\Delta(F,\sigma) = F$-safe and $\Delta_\gamma(F,\sigma_\gamma) = F$-ambiguous. Using the same type of reasoning, it also follows a contradiction, hence the result. □

4.2.2. Algorithm

The proposed algorithm selects a subsystem for which the total cost C_G is minimal (between the supervision cost C_M, the cost C_A to make it accurate and the cost C_D to make it diagnosable for the fault F in component Γ_i). It also returns the requirements *Req* as a set of modifications to perform on the subsystem. The expression $subsystem(\Gamma_i,\ k)$ denotes the set of subsystems composed of k components and containing Γ_i. The function *Monitoring* induces a cost C_M for the monitoring of a subsystem depending on the implementation of the diagnosis architecture like explained in 3.2. The functions *CheckingAccuracy* and *CheckingDiagnosability* are boolean. The first function applies a criterion to determine if the considered subsystem is accurate or not like in Pencolé et al. (2006) and the second one uses an algorithm to check the subsystem diagnosability for a fault F like in Pencolé (2005), Jiang et al. (2001). These algorithms are polynomial in the number of states in $\|\gamma\|$. The function *MakeDiagnosable* uses techniques from sensor placement or other techniques to obtain a diagnosable subsystem γ as described in Section 3.1. The function *MakeAccurate* could rely on the Property 2. It is a simple mean to obtain an accurate subsystem but it is not the only solution, other modifications from Table 1 can also be considered.

```
Input: $F \in \Sigma_{f_i}, \Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$
$C_G^0 \leftarrow \infty$; $k \leftarrow 1$; $Req = \emptyset$
repeat
  $C_G^k \leftarrow \infty$
  for all $\gamma \in subsystem(\Gamma_i,\ k)$ do
    $C_M \leftarrow Monitoring(\gamma)$; $C_D = 0$; $C_A = 0$;
    if $CheckAccuracy(\gamma)$ then
      if $\neg CheckDiagnosability(\gamma, F)$ then
        $(Req, C_D) \leftarrow MakeDiagnosable(\gamma, F)$
      end if
    else
      $(Req, C_A) \leftarrow MakeAccurate(\gamma)$
      if $\neg CheckDiagnosability(\gamma, F)$ then
        $(Req, C_D) \leftarrow MakeDiagnosable(\gamma, F)$
      end if
    end if
    $C_G^\gamma \leftarrow C_M + C_A + C_D$
    $C_G^k \leftarrow \min(C_G^k, C_G^\gamma)$
  end for
  $k \leftarrow k + 1$
until $(C_G^{k-1} \geq C_G^{k-2}) \wedge (k \geq 2) \wedge (k \leq n)$
$C_G \leftarrow C_G^{k-2}$
Output: $\gamma$; $C_G$; $Req$
```

5. RELATED WORK

In the literature, there are many works about fault diagnosis and diagnosability on discrete-event systems. Our framework is based on the classical framework defined in Sampath et al. (1995). The problem of checking diagnosability has been studied for many years and several algorithms have been proposed for centralized monitoring architectures Jiang et al. (2001), Yoo and Lafortune (2002). Our proposal relies on the diagnosability of distributed discrete event systems that has been defined in Pencolé (2005). None of these works propose any design feedback to increase the system diagnosability.

A possible way to provide design requirements for diagnosability is to solve the sensor placement problem. In this type of problem, it is assumed that there always exists a set of modifications of type 1 (see section 3.1) by the addition of sensors to guarantee the diagnosability of the whole system. Moreover, the cost of the monitoring architecture is supposed to be negligible. In Debouk et al. (1999), the problem is to select an optimal subset of available sensors by inferring a set of sensor tests. This inference assumes that the cost of $n + 1$ sensors is always greater than the cost of n sensors (i.e. optimal = minimal) which may be a restrictive assumption from an economical point of view. Jiang et al. (2003) proposes a methodology to select a minimal set of sensors to preserve properties like normality, diagnosability and proves that this problem is NP-hard. In Narasimhan et al. (1998), the problem is defined over a qualitative model and in Torta and Torasso (2007) it is extended by parametrizing some discriminability relations of the system. Here also, optimality is equivalent to minimality. In Spanache et al. (2004), the selection is performed by a genetic algorithm to make a continuous system more diagnosable. The cost of the sensor takes into account the economical issue.

To our best knowledge, there is no work about improving diagnosability in a distributed framework. In our recent work Pencolé (2005), the question of improving diagnosabilty in a distributed discrete event systems has been introduced. The idea is to detect local undiagnosable scenarios that can be used to provide design requirements for the elimination of such scenarios.

6. CONCLUSION AND PERSPECTIVES

This paper defines a framework based on a classical model-based diagnosis formalism in order to extend automatic diagnosability analyses and provide design requirements for a distributed dynamic system. We propose to define the problem as a cost optimization problem where not only the costs about the design of the system but also the costs about the monitoring architecture are taken into account in order to minimize the integration costs of the distributed system. Finally, we claim that the design requirements for diagnosability are closely related to the design requirements for accuracy as this property is a way to isolate a subsystem above which it is possible to design a monitoring architecture that provides a diagnosis as accurate as possible. In this paper, we have presented an algorithm which selects a subsystem for which the total cost C_G is minimal. The solution is not unique but the algorithm can be extended to provide a set of possible subsystems which have a minimal cost. Our perspectives are to develop this methodology in detail by integrating the different methods (diagnosability checkers, sensor placement selectors) and automatically provide design requirements for distributed systems.

REFERENCES

L. Console, C. Picardi, and M. Ribaudo. Process algebra for systems diagnosis. *Artificial Intelligence*, 142:19–51, November 2002.

R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection for failure diagnosis. In *38th Conference on Decision and Control*, pages 4990–4995, Phoenix, Arizona USA, December 1999.

R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *JDEDS: Theory and Application*, 10(1–2):33–86, 2002.

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.

S. Jiang, R. Kumar, and H. E. Garcia. Optimal sensor selection for discrete event systems under partial observation. *IEEE Transactions on Automatic Control*, 48:369–381, March 2003.

G. Lamperti and M. Zanella. *Diagnosis of active systems*. Kluwer Academic Publishers, 2003.

S. Narasimhan, P.J. Mosterman, and G. Biswas. A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems. In *Proceedings of the 9th International Workshop on Principles of Diagnosis*, 1998.

Y. Pencolé. Assistance for the design of a diagnosable component-based system. In *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 549–556, 14-16 Nov 2005.

Y. Pencolé and M.-O. Cordier. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence*, 164:121–170, May 2005.

Y. Pencolé, D. Kamenetsky, and A. Schumann. Towards low-cost diagnosis of component-based systems. In *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process*, Beijing, China, September 2006.

M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event system. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

S. Spanache, T. Escobet, and L. Trav-Massuys. Sensor placement optimisation using genetic algorithms. In *15th International Workshop on Principles of Diagnosis (DX'04)*, pages 179–184, Carcassonne (France), 11-14 Juin 2004.

D. Thorsley and D. Teneketzis. Active acquisition of information for diagnosis of discrete-event systems. In *42th Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, 2004.

G. Torta and P. Torasso. Computation of minimal sensor sets from precompiled disciminability relations. In *18th International Worshop on Principles of Diagnosis (DX'07)*, pages 202–209, Nashville, TN, USA, May 2007.

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