Active Diagnosis of Hybrid Systems Guided by Diagnosability Properties - Application to Autonomous Satellites

Mehdi Bayoudh

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Institut National Polytechnique de Toulouse*
Discipline ou spécialité : *Systèmes embarqués*

Présentée et soutenue par *Mehdi Bayoudh*
Le *4 Février 2009*

Titre : *Active diagnosis of hybrid systems guided by diagnosability properties*
Application to autonomous satellites

JURY
Janan ZAYTOON, Président
Vincent Cocquempot, Rapporteur
Marie-Odile Cordier, Rapporteur
Michael HOFBAUR, Examinateur
Xavier Olive, Examinateur

Ecole doctorale : *École Doctorale Systèmes (EDSYS)*
Unité de recherche : *Groupe de Diagnostic, Supervision et Conduite (LAAS-CNRS)*
Directeur(s) de Thèse : *Louise Travé-Massuyès*
Rapporteurs : *Noms des rapporteurs (s'ils ne font pas partie des membre du jury)*

Université de Toulouse
Institut National Polytechnique de Toulouse
Laboratoire d'Analyse et d'Architecture des Systèmes, LAAS-CNRS

Thèse

présentée et soutenue en vu d'obtenir le grade de Docteur,
spécialité « Systèmes embarqués »

par

Mehdi Bayoudh

Active Diagnosis of Hybrid Systems Guided by Diagnosability Properties

Application to Autonomous Satellites

Thèse soutenue le 04 Février 2009 devant le jury composé de :

M.	Janan Zaytoon	Université de Reims	(Président du jury)
M.	Vincent Cocquempot	Université de Lille	(Rapporteur)
M^{me}	Marie-Odile Cordier	Université Rennes 1	(Rapporteur)
M.	Xavier Olive	Thales Alenia Space	(Examinateur)
M.	Michael Hofbaur	Graz University of Technology	(Examinateur)
M^{me}	Louise Travé-Massuyès	LAAS-CNRS	(Directeur de thèse)

À mes parents

Remerciements

Je tiens à remercier en tout premier lieu Louise Travé-Massuyès, ma directrice de thèse pour sa bonne humeur, sa disponibilité, ses encouragements, son sens de l'écoute, sa présence, tout au long de ces années de thèse.
J'ai toujours admiré sa force de travail, sa rigueur scientifique, sa grande passion pour la recherche, l'ampleur de ses connaissances mais surtout ses qualités humaines, sa modestie, sa patience ...
Louise, qui malgré ses responsabilités innombrables, ses réunions interminables, a toujours su trouver le temps nécessaire pour suivre mes réflexions parfois trop philosophiques, mon avancement souvent non linéaire, mes équations toujours trop compliquées, sa porte de bureau m'était toujours grande ouverte.
Merci Louise pour ces trois années qui ont été vraiment très riches en souvenirs : de nos réunions de travail souvent très vives, de nos discussions bien animées, de nos journées (et parfois soirées !) de travail.
Je remercie également Xavier Olive mon tuteur Thales Alenia Space, pour son grand professionnalisme et ses conseils souvent très pertinents.
Je voudrais remercier mes rapporteurs, Vincent Cocquempot et Marie-Odile Cordier pour l'intérêt qu'ils ont porté à mon travail, les remarques et les suggestions qu'ils m'ont faites et qui m'ont été utiles pour réaliser la version finale de ce manuscrit.
Je souhaite aussi remercier les autres membres de jury, Janan Zaytoon qui m'a fait l'honneur de présider mon jury de thèse et Michael Hofbaur avec qui j'ai eu le plaisir de collaborer et que je tiens à remercier particulièrement pour m'avoir chaleureusement accueilli pendant un mois au sein de son équipe de recherche à l'institut de commande et d'automatique de l'Université de Technologie de Graz en Autriche.
Je souhaite également exprimer ma gratitude à Yannick Pencolé qui m'a fait profiter de sa grande connaissance des systèmes à évènements discrets en répondant à mes diverses questions et qui était toujours prêt à me donner son avis sur mes travaux. Je le remercie donc pour toutes ses remarques toujours très pertinentes et pour son aide précieuse à résoudre mes problèmes avec C++.

Mais cette thèse s'est également inscrite dans un environnement humain particulièrement chaleureux. Je pense à toutes ces personnes grâce à qui je ne me suis jamais senti seul, personnes qui m'ont accompagné au long de ces années de travail. Je conclurai donc en remerciant mes amis : le grand Vincent Albert pour sa joie de vivre, François Armando pour son humour particulier, Fabien Perrot et Nicolas Van Wambeke pour leur précieuse aide sur les problèmes linuxiens, Pauline Ribot et Elodie

Chanthery pour leurs tendres chocolats, Xavier Pucel pour ses blagues non homologuées, Hervé Ressencourt et Siegfried Soldani pour les bons moments passés ensemble, spécialement sur la route de la Peñaranda De Duero, Renaud Pons et Emmanuel Benazera les "anciens", et bien sûr mes amis Tunisiens : Youssef, Manel, Wafa, Mehdi, Walid ...
J'aimerais aussi remercier les membres du groupe DISCO dans leur ensemble ainsi que notre ancienne secrétaire Eliane Dufour, maintenant à la retraite, pour son efficacité, sa disponibilité et pour m'avoir toujours aidé à faire mes réservations de voyage souvent à la dernière minute et à obtenir le bon renseignement alors que je me perds si facilement dans la forêt administrative.
Et je termine enfin en remerciant tous ceux que j'ai pu oublier.

Toulouse, le 10 Mars 2009.

Table of Contents

List of figures

Introduction

Position of the work in the spatial context

These last fifty years have been marked by the revolutionary progress in the domain of space exploration. The race to conquer space has seen its peak during the cold-war with the competition between the USSR and the USA. Nowadays, space missions have become an area of cooperation between countries, notably with the International Space Station (ISS) and the European Space Agency (ESA). During all these years, the technological advancements in several areas, especially computer engineering, electronics and embedded systems have an impact on the development of the know-how and the mastering techniques answering space domain requirements. The long duration of space programs on one hand and the increasing development of modern technologies on the other hand induce high demands on advanced technologies for spacecraft design.
Moreover, space is a harsh environment (the vacuum, temperatures (with big contrasts), radiations, energetic charged particles, degrading chemical agents, orbital debris) for satellite components (electronic components, materials, mechanisms), and failures can be fatal for the mission objective. However, the estrangement between the spatial segment and the ground segment, especially for interplanetary missions whose duration is very long (dozens of years) is a crucial problem for the spacecraft mission success. For such missions the transmission delays between space segment and ground segment (the TM/TC system) are very important. The important delay of the incoming measures translates into an important delay to analyze the health state of the satellite. The important delay of control actions translates into bad reactivity of the ground segment after fault occurrence. The time the sensor telemetry data are received, the diagnosis determined and the control actions sent back, the spacecraft may be lost. Hence, the high demand on autonomy translates into a high need for online monitoring, diagnosis and recovery.
During the last decades, space exploration was disordered by many failing missions :

- *Mars Observer* is the first of a series of NASA planetary missions intended to study the geology and climate of Mars. In august 1993, three days before scheduled Martian orbit insertion, contact with the probe was lost for reasons still not known. Several scenarios for what might have happened during the final moments of Mars Observer were put forward, but none has been confirmed.
- *Beagle 2* was an unsuccessful British landing spacecraft that formed part of the European Space Agency's 2003 Mars Express mission. It is not known for certain whether the lander reached the Martian sur-

face, all contact was lost upon its separation from the Mars Express six days before its scheduled entry into the atmosphere.
- *Phobos program* was an unmanned space mission consisting of two probes launched by the Soviet Union to study Mars and its moons Phobos and Deimos. Phobos 1 operated nominally until an expected communications session on 2 September 1988 failed to occur. The failure of controllers to regain contact with the spacecraft was traced to an error in the software uploaded on 29 August/30 August, which had deactivated the attitude thrusters. By losing its lock on the Sun, the spacecraft could no longer properly orient its solar arrays, thus depleting its batteries.
- *Sakigake (MS-T5)* was Japan's first interplanetary spacecraft. It aimed at demonstrating the performance of the new launch vehicle, test the schemes of the first escape from the Earth gravitation for Japan, observe space plasma and magnetic field in interplanetary space. Contact was lost in January 7, 1999.

These missions could probably have been rescued with the presence of an on-board diagnosis module able to diagnose the faults and decide about the appropriate reconfiguration actions to be performed. On-board diagnosis could have maintained the spacecraft in a safe state, allowing more time for the ground segment to acquire and interpret measurements, and broadcast appropriate control.
In future space missions, the tendency is to bet on autonomy, especially for far-space-missions. The work of this thesis is motivated by the above considerations and aims at defining an active diagnosis approach for a new generation of autonomous satellites. Active diagnosis consists of applying appropriate control inputs able to exhibit a suitable set of symptoms leading to a precise diagnosis of the health status of the spacecraft, hence providing focused information for the reconfiguration module.

Architecture for Autonomy : on-board Active Diagnosis and Reconfiguration

Figure 1 provides the active diagnosis and reconfiguration scheme that we foresee. The proposed architecture is based on the idea that providing a non ambiguous diagnosis is key and given priority as long as it is consistent with the spacecraft safety. Indeed, even the reconfiguration module could accommodate ambiguous faulty situations, the ground segment needs a precise view of the health status of the component to adapt its control strategy.
- The Active Diagnoser performs on-line diagnosis of the system state from available measurements. In the case of non precise diagnosis, additional control inputs are issued in order to exhibit additional symptoms.
- The Planner provides a plan for the system. When the system behavior is normal (no fault), the plan required for mission fulfillment is excused. After a fault is diagnosed, if the diagnosis is precise, a reconfiguration plan is defined and executed, otherwise, a conditional plan is determined for active diagnosis and executed.

- The Hybrid Controller achieves closed loop control. It takes as input incoming measurements and provides suitable discrete and continuous control inputs.
- The Reconfiguration module accommodates to faulty situations by providing a reconfiguration plan. Several reconfiguration strategies can be considered. Reconfiguration can be performed by defining new input commands to drive the system in a degraded mode, by switching on redundant components, etc ... Reconfiguration can be defined on-line or by using a pre-computed reconfiguration-table that associates a suitable plan to every anticipated faulty situation.

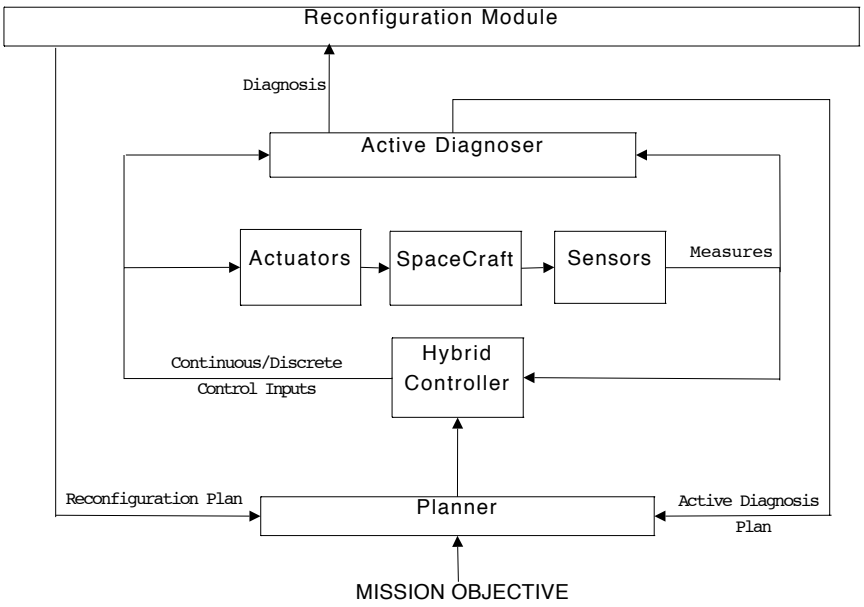

FIG. 1 – *On-board active diagnosis and reconfiguration scheme*

INTRODUCTION OF THE DIAGNOSIS APPROACH

Among existing diagnosis approaches (c.f. Chapters 1 and 2), this work is concerned with model-based approaches. The principle of these approaches is to represent explicitly the physical plant in the form of a model to support the diagnosis reasoning. In model-based approaches, the knowledge about the physical system and the knowledge about the diagnosis task are separated, hence the diagnosis engine is generic and can be used to diagnose different systems, just by changing the system model. Model-based diagnosis is appropriate to take into account different aspects of the system behavior including faulty situations.
This thesis focuses on satellites, that are complex dynamical systems in which the overall physical plant is inherently continuous, but control is often performed by a supervisory controller that imposes discrete switching behavior between several operating modes. Hence a hybrid modeling framework that takes into account both continuous and discrete dynamics is required for the model-based diagnosis approach. The appropriateness of on-board model-based diagnosis for autonomous spacecraft

has been proved with the model-based diagnosis software LIVINGSTONE (c.f. Williams and Nayak (1996)) developed by MIT, JPL and NASA Ames Research Center and used on Deep Space 1. Then, model-based diagnosis was used for space domain applications with the French diagnosis software KOALA Benazera (2003) developed by LAAS-CNRS (Centre National de la Recherche Scientifique) in collaboration with CNES (Centre National des Etudes Spatiales), the monitoring and diagnosis system TRANSCEND Manders *et al.* (February 1999) etc... These approaches can be qualified as *"passive"* in the sense that control and diagnosis are performed separately. Moreover, these approaches do not make use of the diagnosability properties of the system. While, in the autonomy context, the diagnosability property is very important and guarantees that, after the occurrence of a fault, the state of the system can non-ambiguously be determined. Active diagnosis interleaves diagnosis and control and can be considered in order to disambiguate an ambiguous estimated state of the system. It consists of applying additional control inputs to exhibit further symptoms. In this thesis, we propose a hybrid modeling framework and an approach for hybrid system diagnosis which couples discrete event and continuous techniques. The same hybrid model is used to define the diagnosability property for hybrid systems and deriving diagnosability criteria and to support the on-line diagnosis reasoning. The diagnosis approach is extended to active diagnosis guided by the diagnosability properties of the system. The hybrid system is seen as the contribution of two underlying discrete event and continuous systems. The underlying continuous system called *the multimode system* is diagnosed following an extension of the parity space approach. New concepts of *mirror* and *mode signatures* are introduced and capture diagnosis information about the underlying continuous system behavior. Then, signature switches are abstracted in terms of discrete events typifying continuous dynamics and are used to enrich the underlying discrete event system model. From the enriched discrete event model called *the behavior automaton*, we build the diagnoser of the hybrid system to perform on-line diagnosis. It takes as input natural observable discrete events and events issued from the abstraction of the continuous dynamics changes. The definition of diagnosability of multimode systems is introduced based on the new concept of mode signature. By putting together events issued from the abstraction of continuous dynamics and natural discrete events, a prefix closed language is defined and describes the evolution of the hybrid system. Based on this model, the definition and criteria for diagnosability of hybrid systems are provided. It is shown that diagnosability of either the underlying continuous system or the discrete event system are sufficient but not necessary conditions for hybrid diagnosability. Finally, based on the above presented background, a method for performing active diagnosis of hybrid systems is proposed. Starting with an ambiguous belief state, our method calls for diagnosability analysis results to determine a new system configuration in which fault candidates can be discriminated. The control inputs to be applied to the system to drive it into the selected configuration are then determined paying attention to avoid states that could be dangerous for the system. The active diagnosis problem is formulated as a conditional planning problem. The search for active diagnosis actions is guided by the observable

response of the system and by the introduced concept of *controllable path*. From an ambiguous state of the diagnoser the plan defines how to find a controllable path leading to a non-ambiguous state.

Roadmap of the thesis

The thesis starts with a state of the art of model-based diagnosis approaches, diagnosis of hybrid systems, and diagnosability analysis, respectively provided in Chapters 1, 2 and 3. The second part of the thesis presents our active diagnosis approach. First, our formalism for hybrid system modeling as well as our passive diagnosis approach are introduced in Chapter 4. Then, based on this formalism, Chapter 5 presents the underlying diagnosability analysis approach. Next, the diagnosis approach is extended to an active diagnosis guided by diagnosability properties of the hybrid system and provided in Chapter 6. Finally, the proposed diagnosis approach is demonstrated on the Control Attitude System (ACS) of a satellite stabilized by four reaction wheels [1] presented in Chapter 7. This system is diagnosable and consequently diagnosed by means of the passive diagnosis approach (the diagnosis scheme does not call for active diagnosis).

[1]The case study is provided by Thales Alenia Space, France.

PART I

State of the Art on Model-Based Diagnosis and Diagnosability of Hybrid Systems

Chapter 1
Approaches for Model Based Diagnosis

Contents

Diagnosis is concerned with the development of algorithms that are able to determine whether the behavior of a system is normal. If this is not the case, the algorithm should be able to determine, as accurately as possible, which part of the system is failing, and which kind of fault it is undergoing. In the literature, there are several diagnosis approaches developed along theories of Artificial Intelligence, Automatic Control and Statistics.
We can distinguish model-based approaches that require an accurate (analytic, qualitative, logic, ...) model of the system and non model-based approaches, i.e. approaches that do not assume any model and rely only on historic process data.
The main non model-based approaches developed for diagnosis, in the field of Artificial Intelligence are :

- Pattern recognition : the diagnosis is performed by means of a set of indexed observations which are used to identify data classes corresponding to different operating modes (normal and faulty) of the system. In a sense, this approach can be likened to a model-based approach. However, in opposition to the model-based approaches the model is not issued from physical, chemical, etc... considerations, system components and their interactions, but from a data base composed by collected measure samples (c.f. Dubuisson (2001)). There are several approaches of pattern recognition : the structural pattern recognition (c.f. Schalkoff (1992)) that uses the relations between pattern components and numeric (statistic, fuzzy, etc...) pattern recognition that uses probabilistic (or fuzzy) models of the patterns. The last one is the most appropriate to the diagnosis problem.
- Learning approaches : the diagnosis is translated into a learning problem. These approaches assume that the available knowledge of the

system is only given by past and current observations. Then, from these observations a diagnosis system is "learned". The learning aims at searching a set of computable relations between observable input and observable output variables. These relations are used to estimate output variables knowing only input variables. Neuronal and Bayesian Networks can be used to build learning systems (c.f. Schalkoff (1992), Neal (1996)).

In the field of system control and signal-based approaches the main diagnosis approaches are based on physical redundancy, frequency analysis of measurements (c.f. Morita and Okitsu (1990)) and statistical methods (c.f. Bakhache and Nikiforov (2000)).

This thesis is concerned with model-based diagnosis approaches. Model-based diagnosis has been one of the very active research domains in the last decades in both Automatic Control and Artificial Intelligence fields. Several paradigms for qualitative (c.f. Travé-Massuyès and Dague (2003)) and quantitative modeling of physical plants have been proposed and allow one to model complex dynamical systems in order to perform diagnosis. The system model offers the possibility to integrate the physical plant behavior knowledge in the diagnosis module that takes as input on-line measurements provided by the sensoring and monitoring system. Indeed, model-based diagnosis consists of checking the consistency between measurements and the system model. Then, a fault manifests itself and is detected as an inconsistency between the system model and the observations. The use of fault models allows one to determine the fault (identification), by checking the consistency between measurements and models of anticipated faulty behaviors.

It is important to notice that a system model depends on the nature of the system behavior. Hence, we can distinguish :

- Continuous systems, in which the system behavior is described by a continuous state, i.e. by variables that evolve in a continuous domain and describe the evolution of physical phenomena governed by physical laws modeled by algebraic and differential equations.
- Discrete-event systems, in which the system behavior is described by a discrete state, i.e. by discrete variables that evolve in finite discrete domains [1]. These systems can be modeled with logic formulas or finite state machines.
- Hybrid systems that combine both continuous and discrete event dynamics.

Several communities dealing with model-based diagnosis have worked in parallel, developing their own and specific methods that depend on the system modeling framework. We distinguish two main communities, on one hand the Artificial Intelligence community also called *diagnosis from first principles* (DX community) (c.f. Reiter (1987)) that uses symbolic and qualitative models with logic, and on the other hand the Automatic Control community also called *FDI* (Fault Detection and Isolation) community that develops control and statistic decision theories for model-

[1] Notice the difference between discrete systems (the system behavior is described by discrete variables evolving in finite discrete domains) and discrete time representation of continuous systems (the system behavior is described by continuous variables considered at sampling times).

based diagnosis using analytic models and linear algebra (c.f. Chow and Willsky (Jul 1984), Staroswiecki and Comtet-Varga (2001a). Last years have seen bridge works that compare the approaches in the two fields and have established some equivalence between DX and FDI concepts demonstrated in Cordier *et al.* (2004), published in the special issue "*Diagnosis of Complex Systems : Bridging the Methodologies of the FDI and DX Communities*" (c.f. Biswas *et al.* (2004)).

1.1 Fault Detection and Isolation (FDI)

Fault Detection and Isolation algorithms developed by the Automatic Control community rely on quantitative models deduced from the physical rules that govern the system behavior or estimated from input/output data. Quantitative models can be formulated in the temporal domain (state space models with continuous or discrete time representation) or in the frequency domain (transfer functions). The first formulation can be applied to both linear and non linear systems, however, the second formulation is limited to linear systems. A fault is detected by an inconsistency between the model of normal behavior and the observations (detection phase). Then, models of faulty behaviors are used according to the same principle to precisely determine the fault nature (isolation phase). Fault detection and isolation algorithms can be based on analytic redundancy, state estimation (or filtering) and parameter estimation techniques.

– Analytical redundancy : the key idea is to use of analytic redundancy to build testable models, i.e. models that only involve observable variables. Existence of such models is conditioned by redundant information conveyed by the system model. Testable models must be able to reject perturbations (robustness), to detect faults (detectability) and to discriminate them (distinguishability) as presented in Frank (1990). Testable models correspond to Analytic Redundancy Relations (ARRs), that are constraints linking only observable variables (c.f. Chow and Willsky (Jul 1984)). ARR techniques have been developed to deal with linear systems, then they were extended to non linear systems in Staroswiecki and Comtet-Varga (2001a). The system model involves non measurable variables (the state variables) that must be eliminated to obtain the ARRs. The analysis of the existence of ARRs can be made using a structural model of the system in the form of a bipartite graph (c.f. Staroswiecki and Declerck (1989)). Furthermore, structural analysis offers a tool to analyze the detectability and isolability properties of the system as presented in Staroswiecki and Comtet-Varga (2001b). Then, several techniques are proposed to effectively obtain the ARRs analytical expressions, on one hand for linear systems by means of the parity space approach as presented Chow and Willsky (Jul 1984) and on the other hand for non linear systems as presented in Staroswiecki and Comtet-Varga (2001a).
  For linear systems, the parity space approach allows one to eliminate the state variables by projection on the *Parity space*. The extension of this approach to non linear systems is given in Chow and Willsky (Jul 1984). In the absence of unknown inputs (no perturbations), the obtained ARRs involve only observable inputs and outputs, the faults and the noise signals. Hence, they can be directly used to build indicators for fault diagnosis.
  Consider a linear system modeled by its state space and observation equations with discrete time representation :

$$\begin{cases} X(n+1) &= AX(n) + BU(n) + F_x\delta(n) + E_x\epsilon(n) \\ Y(n) &= CX(n) + DU(n) + F_y\delta(n) + E_y\epsilon(n) \end{cases} \tag{1.1}$$

X, U, Y represent the continuous state vector of the system, the input vector and the output vector, of dimensions n_x, n_u and n_y respectively. A, B, C and D are matrices with appropriate dimensions that denote dynamic, input, measure and direct transmission matrices, respectively. n represents the sampling time. δ and ϵ denote the fault and the noise input vectors respectively . F_x, E_x, F_y and E_y are matrices with appropriate dimensions that capture the influence of the fault and the noise, respectively, on state evolution and observations.
State variables are eliminated by iterating the state and the observation equations. The number of iterations defines the order of the parity space and determines the number of consecutive inputs and outputs to be considered to compute the ARRs.
Let p denote the order of the parity space.
$U^p(n) = [U^T(n-p), ..., U^T(n-p+k), ..., U^T(n)]^T$, $Y^p(n) = [Y^T(n-p), ..., Y^T(n-p+k), ..., Y^T(n)]^T$, $\delta^p(n) = [\delta^T(n-p), ..., \delta^T(n-p+k), ..., \delta^T(n)]^T$ and $\epsilon^p(n) = [\epsilon^T(n-p), ..., \epsilon^T(n-p+k), ..., Y^T(n)]^T$ are the input, the output, the fault and the noise vectors respectively considered at the sampling times : $n-p, ..., n-p+k, ..., n$.
The ARRs are decomposed into a so-called computation and evaluation form. The computation form involves only the input and the output vectors. The evaluation form involves only the noise and the fault vectors. The associated residual is obtained by comparing the computation and the evaluation forms. It is equal to 0 if the ARR is satisfied, 1 otherwise.
The computation form is given by :

$$\rho_c(n) = \Omega^p Y^p(n) - \Omega^p L^p U^p(n) \tag{1.2}$$

where :

$$O^p = \begin{pmatrix} C \\ CA \\ ... \\ CA^p \end{pmatrix}, \Omega^p O^p = 0, L^p = \begin{pmatrix} D & 0 & ... & 0 \\ CB & D & ... & ... \\ ... & ... & ... & 0 \\ CA^{(p-1)}B & ... & CB & D \end{pmatrix}$$

The evaluation form is given by :

$$\rho_e(n) = \Omega^p M^p \delta^p(n) + \Omega^p N^p \epsilon^p(n) \tag{1.3}$$

where :

$$M^p = \begin{pmatrix} F_y & 0 & ... & 0 \\ CF_x & F_y & ... & ... \\ ... & ... & ... & 0 \\ CA^{(p-1)}F_x & ... & CF_x & F_y \end{pmatrix}, N^p = \begin{pmatrix} E_y & 0 & ... & 0 \\ CE_x & E_y & ... & ... \\ ... & ... & ... & 0 \\ CA^{(p-1)}B & ... & CE_x & E_y \end{pmatrix}$$

In this thesis, an extension of the parity space approach is used to generate consistency tests for the underlying continuous behavior of hybrid systems. Hence, more details about the parity space techniques are provided in Chapter 4 and in Appendix A.1.

– Observer techniques developed for both linear and non linear systems can also be used for fault detection and isolation (c.f. Magni and Mouyon (1991), Misawa and Hedrick (1989)). The observer gives an estimation of the system state and outputs. Fault detection

is achieved by comparing measured and estimated outputs.
To illustrate the method, consider a continuous linear system modeled by its state space representation with discrete time as shown in equation (1.1). Noise and perturbations are not considered.
The observer is a dynamical system [2], built from the analytic model of the system, that estimates the system state and output. The general form of an observer is defined as follows :

$$\begin{cases} \hat{X}(n+1) &= A\hat{X}(n) + BU(n) + L(\hat{Y}(n) - Y(n)) \\ \hat{Y}(n) &= C\hat{X}(n) + DU(n) \end{cases} \tag{1.4}$$

$\hat{X}$ and $\hat{Y}$ are the estimated state and output respectively. The gain matrix L must be chosen such that the eigenvalues of the matrix $A + LC$ are included in the unit circle. This guarantees the convergence of the state estimation. The estimation error is given as the difference between the real state and the estimated state of the system : $e(n) = \hat{X}(n) - X(n)$. Then, the residual vector is obtained as the difference between the output measurement (provided by the sensors) and the output estimation, $\rho = \hat{Y}(n) - Y(n) = C.e(n)$.
In the same way as for input/output testable models, structural analysis can be used to analyze detectability and isolability properties. The general form of an observer is not suited for the resolution of the isolation problem and it has to be structured to satisfy isolability requirement (c.f. Patton *et al.* (1989)). Moreover, residuals can be optimized to be robust to the perturbations (c.f. Qiu and Gertler (1993)) and to deal with uncertainty (c.f. Adrot *et al.* (1999)).

– Parameter estimation techniques for both linear and non linear systems can be used to develop fault detection and isolation algorithms (c.f. Pouliezos *et al.* (1985)). Fault detection is achieved by comparing estimated parameters to nominal parameters that characterize the normal behavior of the system. Parameters estimation algorithms have to be able to deal with perturbations and uncertainties and to satisfy isolability requirements that rely on identifiability properties (c.f. Grewal and Glover (1976)).

This state of the art does not aim to be exhaustive, only the main ideas of the most known techniques for residual generation are mentioned. Let us notice that in the case of linear systems, the equivalence between observers, parity space and parameter estimation has been established in Patton and Chen (1991).

1.2 Logical diagnosis theory approach (DX)

Over the last 30 years, the DX community has developed an original framework for model-based-diagnosis called *diagnosis from first principles* introduced in Reiter (1987), Hamscher *et al.* (1992). The modeling framework is component-based. The system model describes the system structure (the connections between the system's components) and the system behavior (the behavior of system's components). The behavior description of every

[2] Notice that classically the observer is used in Automatic Control for full state feedback control.

component is issued from physical laws. A diagnosis problem is defined by the system description and a set of observations. The system description and observations are expressed by means of suitable logical formulas (propositional first-order, etc ...). Logic-based diagnosis aims at formally characterizing the set of solutions of a diagnosis problem expressed in term of logic formulas.
Given a system description together with some conflicting observations, the diagnosis problem is set as the one of determining those components of the system which, when assumed to be functioning abnormally, restore the consistency between the observed and the correct system behavior (c.f. Reiter (1987)). In the following we provide the basic definitions and concepts of the classical logical-based diagnosis approach.

Definition 1.1 *The system model is a pair* $(SD, COMPS)$ *where :*
- *SD (System Description) is a set of first order logic formulas that describe the system behavior*
- *COMPS (Components) is a finite set of constants that represent system's components*

To describe the system behavior a specific predicate AB is defined and interpreted to mean abnormal. Given a system component $c \in COMPS$, $AB(c)$ means that the component c is faulty.

Example 1.1 *Let us consider the polybox example shown in Figure 1.1, where* $M1, M2$*, and* $M3$ *are multiplier components,* $A1$ *and* $A2$ *are adder components. The system*

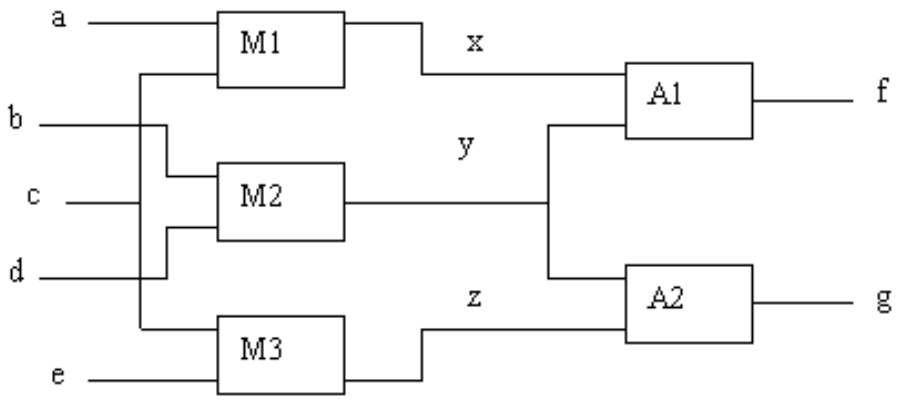

FIG. 1.1 – *The polybox system*

model is given by $(SD, COMPS)$ *with :*
$COMPS = \{A1, A2, M1, M2, M3\}$
$SD = \{ADD(x) \wedge \overline{AB}(x) \Rightarrow Output(x) = Input1(x) + Input2(x)$
$MULT(x) \wedge \overline{AB}(x) \Rightarrow Output(x) = Input1(x) \times Input2(x)$
$ADD(A1), ADD(A2)$
$MULT(M1), MULT(M2), MULT(M3)$
$Output(M1) = Input1(A1), Output(M2) = Input2(A1)$
$Output(M2) = Input1(A2), Output(M3) = Input2(A2)$
$Input1(M1) = Input1(M3)\ \}$

Formulas describing the behavior of the components may also be expressed as constraints and then need a constraint solver to be processed.

Definition 1.2 *Diagnosis problem*
A diagnosis problem is a triple $(SD, COMPS, OBS)$ *where :*
- SD *is the system description*
- $COMPS$ *the set of system's components*
- OBS *the set of observations described by a set of first order formulas*

Definition 1.3 *Fault*
A fault corresponds to a set of components $\Delta \subseteq COMPS$ *described by the formula* $\wedge_{c \in \Delta} AB(c)$

Given a diagnosis problem formulated as a triple $(SD, COMPS, OBS)$, a diagnosis is a conjecture that certain components of the system are behaving abnormally. This conjecture has to be consistent with both the system model and observations. Thus, a diagnosis is given by an assignment of the behavioral mode, AB or $\overline{AB}$, to every component of the system in a way consistent with the observations and the model. Formally :

Definition 1.4 *Diagnosis*
A diagnosis for $(SD, COMPS, OBS)$ *is a set of components* $\Delta \subseteq COMPS$ *such that :* $SD \cup OBS \cup \{AB(c) | c \in \Delta\} \cup \{\overline{AB}(c) | c \in COMPS - \Delta\}$ *is consistent.*

Definition 1.5 *Minimal diagnosis*
A minimal diagnosis *is a diagnosis* Δ *such that* $\forall \Delta' \subset \Delta, \Delta'$*is not a diagnosis.*

The concept of *conflict* is introduced Reiter (1987) and is the key to most implemented diagnosis algorithms.

Definition 1.6 *Conflict*
A conflict for a diagnosis problem $(SD, COMPS, OBS)$ *is a set of components* $C \subseteq COMPS$ *such that* $SD \cup OBS \cup \{\overline{AB}(c)\ c \in C\}$ *is inconsistent.*

Definition 1.7 *Minimal conflict*
A minimal conflict *is a conflict* C *such that* $\forall C' \subset C$, C' *is not a conflict.*

The diagnosis process

The diagnosis is processed in 3 phases :
- Conflict detection : the contradiction between observations and predictions (based on the system model) allows one to detect faults.
- Hypotheses generation : this phase consists of generating hypotheses about the behavior of components (normal or faulty) to resolve all conflicts (to restore the consistency between predictions and observations). A (minimal) diagnosis can be seen as a hitting set of the (minimal) conflicts. In this phase, hitting set computation algorithms (c.f. Reiter (1987), Greiner *et al.* (1989), L.Lin and Jiang (2003)) may be used to determine the possible diagnoses.
- Diagnosis discrimination : this phase consists of determining additional informations to discriminate the possible diagnoses. This is achieved by making new measures when it is possible. Otherwise, the system inputs may be changed in order to put it in a new confi-

guration that exhibits new symptoms (conflicts) (c.f. de Kleer and Williams (1987)).

1.3 Bridge

Both DX and FDI communities have been working in parallel along the lines of Model-Based Diagnosis. In Cordier *et al.* (2004), a formal framework is proposed in order to compare the two approaches and theoretical proof of their equivalence together with necessary and sufficient conditions are provided. A comparison between system modeling, observations, the fault concept, diagnosis definition on both DX and FDI sides is provided as well as links between analytic redundancy relations and conflicts. A track of work has been developed that bridges FDI and DX methods in a synergic way (c.f. Biswas *et al.* (2004)).

Chapter 2
Model Based Diagnosis for Hybrid systems

Contents

Hybrid systems have been the focus of many works in the last ten years. They answer the increasing need to represent systems which exhibit combined continuous and discrete dynamics. Diagnosis techniques developed by the Artificial Intelligence and Automatic Control communities to diagnose continuous and discrete-event systems (c.f. Chapter 1) cannot be directly applied. Hence, specific approaches have been developed. In this chapter, we present the main model-based diagnosis tracks developed for hybrid systems, especially in the field of space vehicles.

2.1 Diagnosis of hybrid systems

The main approaches of model-based diagnosis for hybrid systems have been developed taking benefit of the increasing work developed within the communities of hybrid system modeling and control. Several modeling frameworks have been proposed to represent the behavior of hybrid systems : hybrid automata defined in Henzinger (1996), input/output automata proposed in Lynch *et al.* (1996), hybrid Bond-Graph as proposed in Narasimhan and Biswas (2002), quantized systems presented in Lunze (2000) etc...
Hybrid system research gathers works from artificial intelligence and automatic control. Two classes of models are defined : those proposed by the artificial intelligence community in which the discrete-event aspect is predominant (c.f. Williams and Nayak (1996)) and those proposed by the control community that emphasize continuous aspects (c.f. Hofbaur and Williams (2002), Narasimhan and Biswas (2002)). Diagnosis algorithms supported by these models have been developed using a large spectrum of techniques which often extend existing techniques developed for continuous and discrete-event systems.
This part of the state of the art presents the main directions of work and few illustrative examples. Hybrid diagnosis approaches can be classified in several classes :

- Qualitative approaches are based on a qualitative abstraction of the continuous dynamics. In Lunze (2000), the continuous state is represented in the state space and is only accessible through a quantizer. A qualitative value is associated to every state of the system. The quantizer generates a discrete event whenever the qualitative value of the state changes. The continuous state being quantized, discrete methods are then used. This approach fits with the case when the system state is a signal that cannot be measured quantitatively as well as with some industrial applications for which the faulty behavior is indicated by means of alarm messages, which represent quantized signal values. Williams and Nayak (1996) follows a logical-based diagnosis approach in which the qualitative models are translated into propositional formulas. Measurements are discretized by observation monitors so that observations take their values in a finite domain. In Koutsoukos *et al.* (2001) a fault modeling and diagnosis approach for hybrid systems based on a qualitative representation of the fault hypotheses is presented. Generally, qualitative model-based diagnosis approaches cannot isolate faults manifesting as small variations in the system behavior and their performance is limited by the resolution of the observation monitors.
- Approaches based on hybrid state estimation using Kalman filters : the diagnosis of the hybrid system can be formulated as a hybrid state estimation problem. The use of classical estimation techniques like Kalman filters leads to interesting algorithms with an unified representation of the uncertainty. Indeed, noise and disturbances affecting the continuous models as well as the uncertainty about the transitions between operating modes are represented by means of probability distribution functions. However, they are computationally

very expensive because of the need for tracking multiple models as well as the spontaneous transitions between them like fault transitions that may occur every-time. In Hofbaur and Williams (2002) an approach based on a bank of extended Kalman filters has been presented, where only the most likely trajectories are tracked.

– Approaches based on particle filtering : in Koutsoukos *et al.* (2002) a particle filtering algorithm (that takes into account the interaction between continuous and discrete dynamics) is proposed for hybrid estimation. Autonomous transitions are estimated based on complex transition guards. The transition guard estimation improves the robustness of the hybrid estimation algorithm. In Koller and Lerner (2001) particle filtering has been applied also for a class of hybrid systems modeled by dynamic Bayesian networks for which the autonomous transitions between system modes are defined using conditional probability distributions.
– Approaches using guaranteed set computation techniques : hybrid diagnosis based on concurrent automata has been presented in Benazera *et al.* (2002). The uncertainty on continuous variables and system parameters is modeled in the form of numerical intervals and set computation techniques are interlinked with discrete consistency-based methods for hybrid state estimation.
– Approaches based on coupling qualitative and quantitative models : a model-based diagnosis that combines qualitative and quantitative techniques is presented in McIlraith *et al.* (2000). The hybrid modeling framework allows one to represent continuous behaviors described by Differential and Algebraic Equations (DAEs) as well as discrete transitions that dictate mode switching, modeled by finite state automata, temporal logics and switching functions. Then, the diagnosis task is performed on-line and can be divided into two diagnosis stages :
  - the initial conjecturing of candidate diagnoses.
  - subsequent refinement and tracking to select the most likely diagnoses.
  The diagnosis problem is again formulated as a model selection problem. The aim of the diagnosis task is to find a mathematical model and the associated parameter values that best fit the system data. To address this problem, artificial intelligence techniques for qualitative diagnosis of continuous systems are proposed to generate an initial set of qualitative candidate diagnosis and associated models. To generate these candidates, an abstract model of the dynamical system behavior is constructed as a temporal Bond Graph. This is followed by parameter estimation and model fitting techniques to select the most likely mode and system parameters for candidate models of system behavior, given both past and subsequent observations of system behavior and controller actions. This approach was tested on the *AERCam* robot of NASA (c.f. McIlraith *et al.* (2000)). In Narasimhan and Biswas (2002) a diagnosis approach is proposed to diagnose piecewise linear hybrid dynamical systems. The modeling framework is given by *the Hybrid Bond Graphs* and the used diagnosis techniques combine qualitative reasoning mechanisms with quanti-

tative techniques. Qualitative reasoning supports the fault isolation task that is broken into a hypothesis generation followed by a hypothesis refinement based on the causal Bond Graph model. The quantitative techniques translate into a real-time parameter estimation process using least-squares optimization for fault identification.

2.2 Hybrid Model-Based Diagnosis for Autonomous Spacecraft

One of the most spectacular progress of diagnosis in the space domain is *Livingstone*, developed by the NASA and embedded on-board the probe *Deep Space 1* (c.f. Williams and Nayak (1996)). Livingstone is directly based on the theory of model-based diagnosis. The diagnosis module relies on the propositional logic theory. In the Livingstone formalism, the system dynamics are abstracted in the form of logic formulas (called *qualitative constraints*). The generation of conflicts and diagnosis is formulated as a constraint satisfaction problem solved by an incremental Truth Maintenance System presented in Nayak and Williams (1997). The diagnosis is returned in the form of a set of possible system trajectories.
Livingstone modeling formalism offers the possibility to represent hybrid hardware/software systems by coupling transition system models underlying concurrent reactive languages with qualitative representations of continuous behaviors. Indeed, the system is modeled as a set of concurrent components. The component behavior is described by a probabilistic automaton in which probabilities are linked with every mode transition. The system behavior within a given mode is represented by a set of qualitative constraints expressed with propositional logic formulas. Measurements are discretized by the observation monitors and the observations have finite domains. The benefit of propositional logic theory is the computational efficiency. However, complex continuous dynamics (differential equations) in operating modes cannot be modeled in terms of propositional logic formulas. Furthermore fault detection is limited by the discretization capacity of the observation monitors.
In Benazera (2003) the model-based diagnosis engine *KOALA*[1] has been developed, inspired by Livingstone. The system's components are modeled as hybrid automata and allow one to represent not only qualitative constraints, but also quantitative constraints (algebraic and differential equations). The diagnosis scheme combines interval-based state estimation with consistency-based reasoning. In the hybrid framework of KOALA, the uncertainty is modeled in the form of numerical intervals on continuous variables and probabilities over discrete transition switches. The diagnosis problem is formulated as mode and continuous state estimation. This approach has been demonstrated on the attitude control loop of a standard earth orbiting satellite provided by CNES (Centre National d' Etudes Spatiales).

[1]Work supported by CNES (Centre National des Etudes Spatiales), France and EADS-Astrium, France.

2.3 Position of our work

This thesis introduces the new concept of *Active Diagnosis*, in the sense that the control objectives are temporary modulated in order to diagnose the system state. The diagnosis is so formulated as a joint problem of diagnosis, control and planning. Thereby, a common architecture for diagnosis, planning and control is proposed and improves the autonomy of the system especially in the space domain. In our approach the diagnosis is processed in two stages :
- First stage : the passive diagnosis scheme is performed to estimate the system mode. It takes as input continuous measurements (inputs/outputs) and observable discrete events. The passive diagnosis approach can be compared to other works on diagnosis for hybrid systems.
- Second stage : after an ambiguous passive diagnosis, the active diagnosis process is performed by means of the *active diagnoser* w.r.t controllability and safety considerations. Active diagnosis is formulated as a conditional planning problem. Then, the active diagnosis plan is transmitted to the system controller.

The passive diagnosis approach combines an extension of the parity space approach and the diagnoser approach. Consequently, the system mode is estimated only using observable inputs, output and discrete control inputs (and other observable discrete events). We do not need to estimate the system state to perform diagnosis. Hence, the mode estimation is faster in terms of computation time. In opposition to Hofbaur and Williams (2002) the estimation of the continuous state is not required. Indeed, in Hofbaur and Williams (2002) when more than one system mode is possible, the estimation algorithm tracks all likely modes, hence it is computationally very expensive. Furthermore our approach is generic in the sense that it can be easily extended to take into account different types of constraints (logic formulas, linear or non linear equations ...) as well as different residual generation techniques (to deal with non linear systems for example). Our approach does not deal with probabilities that can be linked with mode transitions. Indeed, probabilities will allow us to define the most likely mode when the system diagnosis is ambiguous. However, in our approach to deal with ambiguity (when the system mode is ambiguous) we perform the second stage of the diagnosis scheme : i.e. we perform active diagnosis in order to disambiguate the system.
In this thesis, the research of the active diagnosis plan is guided by diagnosabilities properties of the system. Work perspective is to use probabilities to improve the research efficiency of the diagnosis plan. In this case, our approach can be extended using probabilistic hybrid automata.

Chapter 3

Frameworks for diagnosability analysis

Contents

The diagnosability of a supervised system is the property that guarantees that after a fault occurrence, the diagnosis module is able to diagnose the fault without ambiguity i.e. the diagnosis module is able to detect the fault and to discriminate it from all the other faults. The definition of the diagnosability property depends mainly on the system model (that must represent faithfully the system behavior), the diagnosis approach adopted to design the diagnosis module and the observation system (the observation acquisition). Diagnosability is properly defined on one hand for discrete-event systems and on the other hand for continuous systems. But there are few equivalent results for hybrid systems.

3.1 Diagnosability of discrete-event systems

The first diagnosability definition of discrete-event systems was provided in Sampath *et al.* (1995) as well as the necessary and sufficient criterion to check diagnosability. The diagnosability checking is based on the diagnoser that is a finite state machine built from the system model. The disadvantage of this method is that the state space of the diagnoser is in the worst case exponential in the cardinality of the state space of the system making the diagnosability checking algorithm very computationally expensive. In Jiang *et al.* (2001) and Yoo and Lafortune (2002) polynomial-time diagnosability verification algorithms have been proposed to reduce this computation problem. The formal verification of diagnosability by means of symbolic model-checking techniques has been proposed in Cimatti *et al.* (2003). All these methods are based on the global representation of the entire system.
On the other hand, approaches to analyze diagnosability in a decentralized way (based on local diagnosers) have been proposed in order to reduce the computational problem. In Pencolé (2004), Pencolé proposes a way to analyze the diagnosability of the system in a decentralized way without the use of the global model. The diagnosability analysis approach is based on local diagnosers built from a distributed model of the system. In Contant *et al.* (2006), an algorithm for diagnosability checking based on a modular representation of the entire system has been proposed and its correctness has been proved.
Extensions of the diagnosability definition have been proposed for stochastic discrete-event systems in Thorsley and Teneketzis (2005), Liu and Qiu (2008) and for fuzzy discrete-event systems in Kilic (2008), corresponding diagnosers and checking algorithms have been provided. In this thesis, we draw one's inspiration from the classical diagnosability definition as proposed for discrete-event systems in Sampath *et al.* (1995) to characterize the diagnosability of hybrid systems.
In the following sections, we provide the basic concepts and definitions originally introduced in Sampath *et al.* (1995) and taken up thereafter by researchers of the discrete-event systems field : Jiang *et al.* (2001), Yoo and Lafortune (2002), Pencolé (2004), Thorsley and Teneketzis (2005), Liu and Qiu (2008). These concepts and definitions are required to formalize our active diagnosis approach and will be recalled later in Chapters 4, 5 and 6.

3.1.1 Modeling framework for discrete-event systems

As classically defined in languages and automata theory (c.f. Hopcroft *et al.* (2000)), a discrete-event system is modeled as a finite state machine $M = (Q, \Sigma, T, q_0)$, where :

- Q is the set of discrete states of the system
- Σ is the set of events
- $T \subseteq (Q \times \Sigma \rightarrow Q)$ is the partial transition function
- q_0 is the initial state

Some of the events in Σ are observable, the rest are non observable. Thus, the event set Σ is partitioned as $\Sigma = \Sigma_{uo} \cup \Sigma_o$, where Σ_{uo} (Σ_o) is the

unobservable (observable) event set. The observable events can be used to model discrete control inputs, discrete sensor readings and communication messages. The unobservable events model the fault occurrences and changes in the system state that are not recorded by the sensors.
The fault occurrence is modeled by a discrete event $f \in \Sigma_F$, where Σ_F models the set of anticipated fault events. Without loss of generality it is assumed that $\Sigma_F \subseteq \Sigma_{uo}$, since an observable fault event is obviously diagnosable. The set of fault events Σ_F is partitioned into disjoint sets corresponding to different fault types, $\Sigma_F = \Sigma_{F_1} \cup \Sigma_{F_2} \cup ... \cup \Sigma_{F_m}$ where m is the number of different fault types in the system.
The behavior of the discrete-event system is described by a string of events (called *trajectory*) : $s = e_1.e_2...e_k$, where $e_i \in \Sigma$, $i \in 1..k$. The set of all possible trajectories forms a prefix-closed language (c.f. definition 3.1) over the event alphabet Σ, denoted $L(M)$. $L(M)$ is a subset of Σ^*, where Σ^* denotes the set of all finite strings of elements of Σ (including the empty string ϵ) termed *the Kleene-closure* of the set Σ (c.f. Ramadge and Wonham (1989)).

Definition 3.1 *Prefix-closed language*
A string u is a prefix of a string $v \in \Sigma^*$ *if for some* $w \in \Sigma^*$, $v = uw$.
The prefix closure of a language $L \subset \Sigma^*$ *is defined to be the language :* $\bar{L} = \{u : uv \in L \text{ for some } v \in \Sigma^*\}$.
A language L is prefix-closed if $\bar{L} = L$

Notice that, if v is a possible trajectory of the discrete-event system described by the language $L(M)$ then clearly so are all the prefixes of v. Consequently, L(M) is a prefix-closed language.
Generally the following assumptions are made about the discrete-event system M :

- The language L(M) is life i.e. there is a transition defined at every state $q \in Q$.
- There does not exist in M any unobservable cycle (i.e cycles containing unobservable events only).

The liveness assumption of $L(M)$ is made for the sake of simplicity. Otherwise with slight modifications all the main results hold true when this assumption is relaxed. In the community of discrete-event systems, the diagnosis consists in the deduction of unobservable fault events from the observable strings of events generated by the system (c.f. Sampath *et al.* (1995)). The absence of non observable cycles guarantees that observations occur with some regularity.

3.1.2 The diagnoser construction

The aim of diagnosis is to make inferences about past occurrences of faults on the basis of the observed events. In order to solve this problem the system model is converted into a deterministic finite state machine called the *diagnoser* built from the system model as explained in Sampath *et al.* (1995).
In order to explain the diagnoser construction, we recall the following definitions and notations.

First, we define a set of fault labels $\Delta_f = \{F_1, F_2, ..., F_m\}$, where m is the number of different fault types in the system. The set of possible fault labels is defined as $\Delta = 2^{\Delta_f}$. Notice that the empty-set label $\emptyset \in \Delta$ should be interpreted as representing the normal behavior of the system. A label of the form $\{F_i, F_j\}$ should be interpreted to mean that at least one fault of type i and at least one fault of type j have occurred.
Given $s \in \Sigma^*$ a string of events, "$\Sigma_{F_i} \in s$" should be interpreted as at least one fault event of type i belongs to s.
Let s_f denote the final event of a string s and $L(M, q)$ the set of all strings that originate from state $q \in Q$. We define :
$L_o(M, q) = \{s \in L(M, q) \mid s = u\sigma, u \in \Sigma_{uo}^*, \sigma \in \Sigma_o\}$
and $L_\sigma(M, q) = \{s \in L_o(M, q) \mid s_f = \sigma\}$
$L_o(M, q)$ denotes the set of all strings that originate from the state q and end at the first observable event. $L_\sigma(M, q)$ denote those strings in $L_o(M, q)$ that end at the particular observable event σ.
We define $Q_o = \{q_0\} \cup \{q \in Q, \exists(q', \sigma) \in Q \times \Sigma_o$ such that $T(q', \sigma) = q\}$ the set of observable states.
We define the label propagation function $LP : Q_o \times \Delta \times \Sigma^* \rightarrow \Delta$ as :

$$LP(q, l, s) = \begin{cases} \emptyset & \text{if } l = \emptyset \text{ and } \forall i, \Sigma_{F_i} \notin s \\ \{F_i | F_i \in l\} \cup \{F_i | \Sigma_{F_i} \in s\} & \text{otherwise} \end{cases}$$

The diagnoser for an automaton $M = (Q, \Sigma, T, q_0)$ is a deterministic finite state machine $Diag(M) = (Q_D, \Sigma_D, T_D, q_{D_0})$ with :

- $q_{D_0} = \{(q_0, \{\emptyset\})\}$ is the initial state of the diagnoser (we assume that the system M is normal to start with).
- $\Sigma_D = \Sigma_o$ is the set of observable events of the system.
- $Q_D \subseteq 2^{Q_o \times \Delta}$ is the set of states of the diagnoser (states reachable from q_{D_0} under T_D). The states of the diagnoser provide the set of diagnosis candidates as a set of couples whose first element refers to the state of the original system and the second is a label providing the set of faults on the path leading to this state. An element $q_D \in Q_D$ is a set of the form $q_D = \{(q_1, l_1), (q_2, l_2), ..., (q_n, l_n)\}$, where $q_i \in Q_o$ and $l_i \in \Delta$.
- $T_D \subseteq (Q_D \times \Sigma_o \rightarrow Q_D)$ is the partial transition function of the diagnoser defined as follows :
  $T_D(q_D, \sigma) = \bigcup_{(q,l) \in q_D} \bigcup_{s \in L_\sigma(M,q)} \{(T(q, s), LP(q, l, s))\}$

The diagnoser is built off-line from the system model and used for on-line diagnosis and monitoring using observable discrete events. Furthermore, the diagnoser can be used to check the diagnosability property of the system. The diagnoser can be used off-line as a design assistant in order to design diagnosable systems. Diagnosability of discrete-event systems is introduced in the following section.

3.1.3 The diagnosability definition

A discrete-event system is diagnosable if the associated prefix-closed language over the alphabet of events, generated by the automaton model is diagnosable. The classic diagnosability definition for discrete-event systems has been provided in Sampath *et al.* (1995). The system is said to

be diagnosable if and only if all the anticipated faults are diagnosable. Roughly speaking, a fault f is diagnosable if and only if its occurrence is always followed by a finite observable sequence of events that allows one to diagnose f with certainty. Formally :

Definition 3.2 *The discrete-event system is diagnosable if* $\forall f \in \Sigma_F$, $\exists n \in \mathbb{N}$ *such that :*
$\forall s_F t$ *a string of events (or trajectory), such that* s_F *ends with the occurrence of* f*, and* t *is a continuation of* s_F *:*
$||t|| \geq n \Rightarrow (\forall s \in L(M) : P_{\Sigma_o}(s) = P_{\Sigma_o}(s_F t) \Rightarrow f$ *occurs in* $s)$
where P_{Σ_o} *is the projection operator on the set of observable events.*

Definition 3.3 *Uncertain state*
Given a diagnoser state $q_d \in Q_D$*, this state is* F_i*-uncertain if* F_i *does not belong to all the labels of* q_D*, whereas* F_i *belongs to at least one label of* q_D*. Formally : a state* $q_D \in Q_D$ *is* F_i*-uncertain if* $\exists (q,l),(q',l') \in q_D$*, such that* $F_i \in l$ *and* $F_i \notin l'$*.*

Definition 3.4 *Indeterminate cycle*
An F_i*-indeterminate cycle in* $Diag(M)$ *is a cycle composed of* F_i*-uncertain states for which there exist two corresponding cycles in* M *: one involves only states that carry and the other involves states that does not carry, the fault label* F_i *in their labels in the cycle in* $Diag(M)$*.*

Theorem 3.1 *Sufficient and necessary condition for diagnosability*
The system M *is not diagnosable if and only if the associated diagnoser* $Diag(M)$ *contains an* F_i*-indeterminate cycle.*

Example 3.1 *We illustrate by a simple example the construction of the diagnoser. Consider the discrete-event system shown in Figure 3.1 (left). Here,* o_1, o_2 *are observable events,* uo *is an unobservable event while* f_1 *and* f_2 *represent fault events. Let* q_0 *be the*

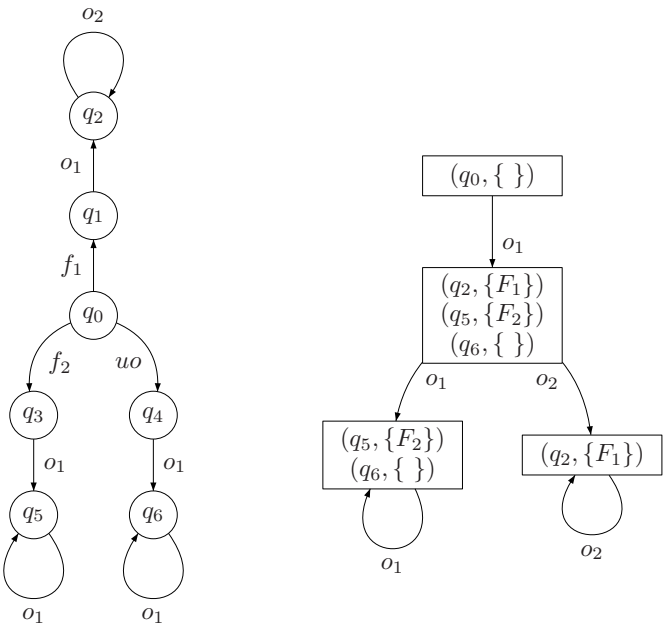

FIG. 3.1 – *The example of a discrete-event system and its associated diagnoser*

initial (normal) state of the system. The set of fault events is partitioned into : $\{f_1\} \cup \{f_2\}$ *that model two different types of faults. The associated diagnoser is provided in Figure 3.1 (right).* F_1 *and* F_2 *denote the fault labels associated to fault events* f_1 *and* f_2 *respectively. In this example, the empty-set label is denoted "*$\{\ \}$*".* $\{(q_2, \{F_1\}), (q_5, \{F_2\}), (q_6, \{\ \})\}$ *is an* F_1 *(*F_2*) -uncertain state of the diagnoser,* $\{(q_5, \{F_2\}), (q_6, \{\ \})\}$ *is an* F_2*-uncertain state. The loop* o_1 *over the* F_2*-uncertain state* $\{(q_5, \{F_2\}), (q_6, \{\ \})\}$ *represents an* F_2*-indeterminate cycle, hence the fault* F_2 *is not diagnosable. However, there is no* F_1*-indeterminate cycle in the diagnoser, hence the fault* F_1 *is diagnosable.*

A relaxed diagnosability definition has been proposed in Sampath *et al.* (1995) and termed *I-diagnosability*. This definition requires the diagnosability to hold not for all trajectories containing the fault but only for those in which the fault event is followed by some associated observable events called *indicators*. Formally :
For every fault event $f \in \Sigma_F$ we associate a set of indicator events. Let $\Sigma_I \subseteq \Sigma_o$ denote the set of all indicator events and let $I_f : \Sigma_F \rightarrow 2^{\Sigma_I}$ denote the indicator map.

Definition 3.5 *I-Diagnosability*
The discrete-event system is I-diagnosable if $\forall f \in \Sigma_F$, $\exists n \in \mathbb{N}$ *such as :*
$\forall s_F t$ *a string of events (or trajectory) such that* s_F *ends with the occurrence of* f*, and* t *is a continuation of* s_F *such that* $I_f(f) \subseteq P_{\Sigma_I}(t)$ *:*
$||t|| \geq n \Rightarrow (\forall\ s \in L(M) : P_{\Sigma_o}(s) = P_{\Sigma_o}(s_F t) \Rightarrow f$ *occurs in* $s)$
where P_{Σ_o} *is the projection operator on the set of observable events and* P_{Σ_I} *is the projection operator on the set of indicator events.*

The corresponding diagnoser condition then proves that the presence of indeterminate cycles *following the indicator events* make the system non I-diagnosable. In the active diagnosis context, the set of indicator events associated to a given fault can be used to take into account the active diagnosis commands that have to be performed to diagnose the system. Hence, the diagnosability definition can be relaxed w.r.t to active diagnosis. This will be discussed in Chapter 6.

3.2 Diagnosability of Continuous Systems

In the FDI community, diagnosability of continuous systems is formulated in terms of fault detectability and isolability provided in Chen and Patton (1994), Nyberg (2002), Travé-Massuyès *et al.* (2006). In Basseville *et al.* (2001) a survey of the several definitions of fault detectability and isolability is provided, in which two types of definitions are distinguished : intrinsic and performance-based (c.f. Basseville and Nikiforov (1993)) definitions. Classically, detectability and isolability are defined as intrinsic properties of the system without any reference to a particular FDI algorithm (c.f. Frisk *et al.* (2003)). This might be compared to observability and controllability, which are defined without any reference to any particular observer or controller. Detectability and isolability definitions are based on the concept of fault signature associated to every anticipated fault. The signatures are generally gathered (c.f. Gertler (1998)) by means of a structure matrix (called *the fault signature matrix*) that expresses the cause-effect

relationships between faults/disturbances as inputs and residuals as outputs. Each column of the matrix represents a fault/disturbance and each row a Boolean residual (c.f. Chapter 1) of the system. A "1" in the intersection means that the fault/disturbance may affect the residual while a "0" means it does not.

Example 3.2 *Let us consider a system with three faults : F_1, F_2 and F_3, three residuals r_1, r_2 and r_3, and the fault signature matrix defined as follows :*

	F_1	F_2	F_3
r_1	1	1	0
r_2	1	1	1
r_3	1	0	1

This means that F_1 may affect all residuals, F_2 may affect r_1 and r_2 and F_3 may affect r_2 and r_3.

The fault signature of a fault F_i is the i^{th} column of the fault signature matrix.

Definition 3.6 *Detectability*
A fault (or a disturbance) is non detectable if its corresponding column in the fault signature matrix contains only "0" elements.

Definition 3.7 *Isolability*
Two faults (or disturbances) F_i and F_j are isolable if they are detectable and their two corresponding columns in the fault signature matrix are different.

Let us then notice that in Example 3.2 faults F_1, F_2 and F_3 are detectable and isolable.
But the signature can also be defined in terms of fault-to-output transfer functions or in terms of the different subspaces in which the output data may live when the system is subject to the different faults. In Basseville *et al.* (2001) it is shown that this definition can also be defined in terms of the amount information about the fault contained in the observed data or in terms of a distance between the normal and the faulty system.
In an opposite way, the second approach is to define the detectability and isolability properties with explicit reference to a particular FDI algorithm taking into account its performances i.e. performance-based definitions are built on indexes of performance of FDI algorithms (c.f. Basseville and Nikiforov (1993)).

3.3 Diagnosability of Hybrid Systems

As mentioned in the introduction of the chapter, the diagnosability definition depends mainly on the used modeling framework as well as on the observation system. Referring to hybrid systems some work directions have been given to characterize the diagnosability properties. In Biswas *et al.* (2006), the used formalism is based on Real Time Hybrid Systems (RTHS). In this formalism the behavior of the hybrid system is described by the notion of *trace* which is a sequence of transitions. Transitions capture the change of both continuous and discrete variables of the system

during mode change. The set of all traces generated by the system model builds up the system language and an algorithm is proposed for the diagnoser construction. The classical necessary and sufficient condition of discrete-event system diagnosability from Sampath *et al.* (1995) is lightly modified because it does not take into account that an uncertain cycle cannot be infinitely crossed due to physical considerations of the underlying continuous behavior. To capture this continuous feature, the necessary and sufficient condition is expressed in terms of reachability. It has the advantage that the diagnosability condition can be checked over the diagnoser without having to refer back to the original system model. However, the reachability analysis can be complex.
In our approach presented in Bayoudh *et al.* (2008a), the trajectory concept defined for hybrid system is equivalent to the notion of trace defined in Biswas *et al.* (2006). However the trajectory is composed by two types of events (our language is heterogeneous) : the naturally discrete events and new events added to capture the continuous dynamics. This allows us to explicitly exhibit discrete-event and continuous aspects of the hybrid behavior.
In Fourlas *et al.* (July 2002) authors use the Hybrid Input/Ouput Automata (HIOA) formalism. The hybrid behavior is described by an *hybrid execution* which is an alternating sequence of continuous evolutions called *trajectories* and actions. This feature is similar to our formalism in which the hybrid behavior is described by alternating events capturing continuous dynamics and "natural" discrete events. The *visible* behavior of the system is described by *the hybrid trace* obtained by the projection on observable variables. Observations are only achieved by the measurement of transition guards (linking continuous variables). This is different from our approach in which we have two kinds of observations : observable "natural" discrete events and observable continuous variables abstracted in terms of observable discrete events. The necessary and sufficient condition for diagnosability proposed by Fourlas *et al.* (July 2002) requires the measurability of transitions guards (even for the guards of fault transitions), and as a consequence, the observability of the continuous system in a given mode. These hypotheses are very restrictive. Furthermore, discrete observations, like "observable" input actions (that can be useful for diagnosis) are not benefited for the diagnosability analysis. On the contrary, in our approach, we use both observable discrete-event and continuous dynamics, consequently, our diagnosability conditions (c.f. Bayoudh *et al.* (2008a)) do not require observability of the continuous system within a given mode.
In Cocquempot *et al.* (2004), the definition of discernability of a hybrid system is given and corresponds to our mutual diagnosability definition provided in Bayoudh *et al.* (2008a). Actually, Cocquempot *et al.* (2004) just considers multimode continuous systems and ignores the discrete event dynamics. Restricting the comparison to multimode systems, our work complete the diagnosability analysis by introducing the important property of *3rd diagnosability*. Our framework proposed to hybrid system diagnosability analysis is detailed in the Chapter 5 of the thesis.

PART II

Active Diagnosis for Hybrid Systems Guided by Diagnosability Properties

Chapter 4

Coupling Continuous and Discrete-Event Techniques for Hybrid Diagnosis

Contents

The use of embedded electronic controllers in physical processes, is increasing, and lead to complex systems that mix both discrete and continuous behaviors. As a consequence, the overall system is hybrid and a hybrid modeling framework is hence needed to take into account continuous and discrete dynamics. On the other hand, the high demands on

performance and availability for such systems, translates itself into a mandatory need for on-line fault detection and diagnosis.
In this chapter we introduce our modeling framework based on hybrid automata (c.f. Henzinger (1996)). In addition to the classic hybrid automata representation, we propose to model the two continuous and discrete-event behaviors by two underlying systems, thus the behavior of the hybrid system is seen as the contribution of both of them. Then, supported by this framework, we introduce the first stage of our diagnosis scheme that corresponds to the passive diagnosis approach, (active diagnosis will be considered in Chapter 6). In our framework, the diagnosis problem is formulated as a mode tracking problem. The tracked modes returned by the passive diagnosis scheme can be a nominal mode, a faulty mode or a set of such modes (in this case, the state of the system is *ambiguous* and active diagnosis is required). The diagnosis scheme is based on the hybrid model. The decomposition of the model into two underlying systems allows us to combine both discrete-event and continuous diagnosis techniques in the same framework. Indeed, the parity space and the diagnoser approaches are combined.

4.1 Hybrid System Modeling

The hybrid system evolves between several operating modes that model both nominal, anticipated faulty continuous behaviors as well as degraded modes modeling reduced operation of the system after fault occurrence yet ensuring the minimal functionalities required for system survival. These latter modes are included in the anticipated faulty modes and model the behavior of the system after the reconfiguration actions. Transitions between nominal modes can be spontaneous or controlled. Transitions between faulty modes can be spontaneous (in the case of multiple faults) or controlled (in the case of a reconfiguration action aiming at driving the system from a faulty mode to a degraded mode). Finally, the transition between a nominal and a faulty mode is spontaneous and modeled by an uncontrollable unobservable fault event. The behavior of the system in an operating mode is modeled by an associated discrete state in the hybrid automaton and operating mode changes are modeled by corresponding discrete transitions labeled by associated discrete-events.
Formally, as mentioned in Henzinger (1996) and Bayoudh *et al.* (2008b), a hybrid system is described by a hybrid automaton defined as a tuple $S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0))$, where :

- ζ is the set of continuous variables, which includes observable and non observable variables. The set of observable variables is denoted by ζ_{OBS} [1].
- Q is the set of discrete system states. Each state $q_i \in Q$ represents a behavioral mode of the system. It includes nominal and anticipated fault modes. The *unknown mode* defined in Hofbaur and Williams (2004), can be added to model all the non anticipated faulty situations.
- Σ is the set of events that correspond to discrete control inputs, spontaneous mode changes and fault occurrences. Events corresponding to spontaneous mode changes are triggered upon guards that depend on continuous variables.
  $\Sigma_o \subseteq \Sigma$ is the set of observable events.
  $\Sigma_{uo} \subseteq \Sigma$ is the set of unobservable events.
  $\Sigma = \Sigma_{uo} \cup \Sigma_o$
- $T \subseteq Q \times \Sigma \rightarrow Q$ is the partial transition function.
- C is the set of system constraints linking continuous variables. It contains differential and algebraic equations modeling the continuous behavior of the system. The set of constraints associated to a mode $q_i \in Q$ is denoted C_i.
- $(\zeta_0, q_0) \in \zeta \times Q$ is the initial condition of the hybrid system.

The occurrence of a fault F_i is modeled by a discrete event $f_i \in \Sigma_F$, where Σ_F models the set of anticipated fault events. Without loss of generality it is assumed that $\Sigma_F \subseteq \Sigma_{uo}$, since an observable fault event is obviously diagnosable.
As mentioned before, the behavior of the hybrid system is seen as the

[1] We assume that the set of system observable variables is the same in all system modes. This assumption is generally verified when the set of system's sensors is permanent, and hence does not depend on the system mode.

contribution of an underlying discrete-event system and an underlying continuous system.

4.1.1 The underlying discrete-event system

The discrete part of the hybrid automaton is modeled as a discrete automaton denoted $M = (Q, \Sigma, T, q_0)$ that describes the discrete dynamics of the system, i.e. the possible transitions between operating modes of the system and their associated events.

4.1.2 The underlying continuous system

The continuous behavior of the hybrid system is modeled by an underlying continuous system denoted $\Xi = (\zeta, Q, C, \zeta_0)$ that describes the whole continuous behavior of the system. Notice that transitions between modes are implicit and consequently not constrained in any way. We hence call this system *the multimode system*.
The underlying continuous behavior in each mode q_i is modeled by a set of constraints C_i. The different modeling frameworks proposed for continuous systems can be used to describe the continuous behavior in every operating mode. In this work, to illustrate our approach and without loss of generality, we deal with linear systems, modeled in the state space by the evolution and the observation equations. Under this assumption, the continuous part of the hybrid automaton is given by the continuous models associated to every mode q_i in the following form :

$$\begin{cases} X_i(n+1) &= A_i X_i(n) + B_i U(n) + E_{x_i}\epsilon(n) \\ Y(n) &= C_i X_i(n) + D_i U(n) + E_{y_i}\epsilon(n) \end{cases} \tag{4.1}$$

$X_i(n)$, $U(n)$, $Y(n)$ and $\epsilon(n)$ are : the state, the input, the output and the noise vectors of dimension n_{x_i}, n_u , n_y and n_ϵ respectively, considered at the sampling time nT_s.
Hence ζ_{OBS} is composed by input and output variables and ζ is composed by all input, output and state variables.
T_s is the sampling period. A_i, B_i, C_i and D_i are constant matrices of appropriate dimensions that denote dynamic, input, measure and direct transmission matrices, respectively. E_{x_i} and E_{y_i} are constant matrices of appropriate dimensions that capture the influence of the noise on state evolution and observations, respectively.
Notice that we do not require to represent the fault vector in the state equations because in our approach we associate a faulty mode and model to every anticipated fault.

4.1.3 Illustrative example : the underlying CS and DES

To illustrate our approach, we introduce the example of a dynamic hybrid system whose underlying discrete part is described by the automaton of Figure 4.1. o_1 and o_2 are observable events (example : discrete control inputs, etc...), f_1 is a fault event and uo_1 is an unobservable event that model a spontaneous transitions. Modes q_1, q_2, q_3 and q_4 represent operating modes of the system : q_1, q_3 and q_4 are nominal, q_2 is an anticipated

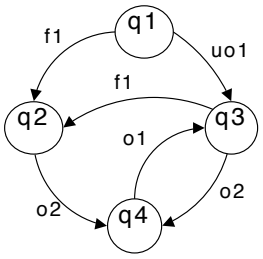

FIG. 4.1 – *The underlying discrete-event system*

faulty mode linked with the fault event f_1. The underlying continuous behavior is given by the state space model of every mode $q_i, i \in 1..4$. In this example, without loss of generality and for sake of simplicity, we consider no noise and no disturbance.

$$\begin{cases} X_i(n+1) &= A_i X_i(n) + B_i U(n) \\ Y(n) &= C_i X_i(n) + D_i U(n) \end{cases} \tag{4.2}$$

where

$A_1 = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix}$, $A_2 = \begin{pmatrix} -0.5 & 4 & 0 \\ 0 & 0.6 & 0 \\ 6 & 0 & 0.8 \end{pmatrix}$, $A_3 = \begin{pmatrix} 0.3 & -0.3 & 0 \\ 0 & 0.6 & 0 \\ -0.3 & 0 & 0.9 \end{pmatrix}$,

$A_4 = \begin{pmatrix} 0.6 & -0.3 & 0 \\ 0.3 & 0.6 & 0 \\ -0.6 & 0 & 0.9 \end{pmatrix}$

$B_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $B_2 = B_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $B_4 = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}$

$C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, $C_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $C_3 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, $C_4 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$

$D_1 = D_2 = D_3 = D_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

This example will be reconsidered further on in this chapter and in Chapter 5 to illustrate the introduced concepts.

4.2 The hybrid system diagnosis approach

The proposed model-based diagnosis scheme is achieved through the following steps (c.f. Bayoudh *et al.* (2008b)) :

- diagnose the underlying continuous system.
- abstract the continuous diagnosis knowledge in terms of discrete events (c.f. Bayoudh *et al.* (2006; 2008a)).
- enrich the underlying discrete-event system by discrete events issued from the abstraction of the continuous dynamics.
- apply the diagnoser approach to the resulting enriched discrete-event system.

Diagnosis (mode tracking) is achieved on-line by the diagnoser approach using observable discrete events and continuous measurements.
Now, in the following sections, the diagnosis problem is specified for hybrid systems and the steps of the diagnosis scheme are developed.

4.2.1 The hybrid system diagnosis problem

Supported by the presented hybrid modeling framework, model-based diagnosis consists in checking the consistency between the system model and incoming measurements (observable continuous variables and observable discrete events). Since anticipated faults are modeled by fault modes with associating faulty behavior models, the diagnosis is then formulated as a mode tracking problem. Diagnosing the hybrid system consists in determining the current state(s) of the underlying discrete-event system which is consistent with incoming measurements i.e. estimating the hybrid system mode.

4.2.2 Diagnosing the underlying multimode system

To check the consistency between the system model and the observations, a set of consistency indicators is linked with every operating mode $q_i \in Q$. A set of constraints C_{obs_i} that involve only observable continuous variables is associated to each mode q_i. Constraints of C_{obs_i} are determined by eliminating non observable variables in the constraints belonging to the set C_i. The constraints are then evaluated on observable variables. Constraints of C_{obs_i} are satisfied when the system evolves in mode q_i. A consistency indicator (the *residual*) is associated to every constraint $C^k_{obs_i} \in C_{obs_i}$ and denoted r_{ik}. The residual is a boolean indicator. It is zero when the constraint $C^k_{obs_i}$ is satisfied, otherwise it is equal to 1.

4.2.3 The extension of the parity space approach to multimode system diagnosis

Following the parity space approach, consistency tests take the form of a set of Analytical Redundancy Relations (ARRs) by eliminating non observable variables (c.f. Chapter 1). For every mode $q_i \in Q$, the set C_{obs_i} is composed by ARRs determined from the system model given by constraints of C_i. Hence a constraint $C^j_{obs_i}$ is an analytic redundancy relation denoted ARR_{ij} and the associated residual is r_{ij} defined as follows :

$$r_{ij} = \begin{cases} 0 & \text{when } ARR_{ij} \text{ is satisfied} \\ 1 & \text{otherwise} \end{cases}$$

$j = 1, ..., N_{r(q_i)}$, where $N_r(q_i)$ is the number of associated residuals/ARRs. The parity space approach is extended to multimode systems and provides analytic redundancy relations that relate the continuous inputs with the observable continuous outputs over a time-window of length $p_i + 1$. Selecting p_i appropriately (typically $p_i \leq n_x$) allows us to eliminate any dependency upon the system state X_i. This procedure can be summarized for a given mode q_i as follows :
Given a vector V, let us denote by V^p the vector obtained by the concatenation of the vector values at every sampling instant $(n - p + k)$, $0 \leq k \leq p$, for a given order p. Hence $V^p(n) = [V^T(n-p), ..., V^T(n-p+k), ..., V^T(n)]^T$.
By iterating state-evolution and observation equations 4.1, we obtain :

$$Y^{p_i}(n) = O_i^{p_i} X_i(n - p_i) + L_i^{p_i}(A_i, B_i, C_i, D_i)U^{p_i} + L_i^{p_i}(A_i, E_{x_i}, C_i, E_{y_i})\epsilon^{p_i}(n) \tag{4.3}$$

$$\text{with : } L_i^{p_i}(M_i, N_i, P_i, Q_i) = \begin{pmatrix} Q_i & 0 & ... & 0 \\ P_i N_i & Q_i & ... & ... \\ ... & ... & ... & 0 \\ P_i M_i^{(p_i-1)} N_i & ... & P_i N_i & Q_i \end{pmatrix}$$

$$O_i^{p_i} = \begin{pmatrix} C_i \\ C_i A_i \\ ... \\ C_i A_i^{p_i} \end{pmatrix}$$

Notice that for a sufficiently high order $p_i \leq n_x$, there always exists a matrix $\Omega_i^{p_i}$ that is orthogonal to the matrix $O_i^{p_i}$, i.e. $\Omega_i^{p_i}.O_i^{p_i} = 0$ (the proof is provided in appendix A.1). So, we can eliminate the state $X_i(n - p_i)$ in equation 4.3 through left-hand multiplication with $\Omega_i^{p_i}$.
Hence we obtain the analytic redundancy relations that can be decomposed into a computational and an evaluation form denoted $\rho_{c_i}^{p_i}$ and $\rho_{e_i}^{p_i}$ respectively and given as follows :

$$\rho_{c_i}^{p_i}(n) = \Omega_i^{p_i} Y^{p_i}(n) - \Omega_i^{p_i} L_i^{p_i}(A_i, B_i, C_i, D_i) U^{p_i}(n) \tag{4.4}$$

$$\rho_{e_i}^{p_i}(n) = \Omega_i^{p_i} L^{p_i}(A_i, E_{x_i}, C_i, E_{y_i}) \epsilon^{p_i}(n) \tag{4.5}$$

The Boolean-residual vector of the mode q_i is denoted $R^{q_i} = [r_{i1}, r_{i2}, ..., r_{iN_r(q_i)}]^T$ and obtained by checking the consistency between computational and evaluation forms.
Notice that in a noise-free environment, the evaluation form of the residuals is null : $\rho_{e_i}^{p_i}(n) = 0, \forall n \in \mathbb{N}$.
In the multimode framework, the set of ARRs linked with each functional system mode is generally different, although some ARRs may be shared.

Illustrative example : the ARRs computation

Now, let us take again the example of Figure 4.1. The *optimal* parity space order[2] is computed for every mode of the system and it is equal to 1, i.e. the computational form is calculated from the continuous observable variables U and Y, at time $n-1$ and n, and given as follows :

$$- \rho_{c_1}^1(n) = \begin{pmatrix} -0.5715 & 0.0471 & 0.8165 & -0.0673 \\ -0.0471 & -0.5715 & 0.0673 & 0.8165 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$$
$$+ \begin{pmatrix} -0.1776 & -0.8165 \\ -0.8367 & -0.0673 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$$

$$- \rho_{c_2}^1(n) = \begin{pmatrix} -0.1278 & -0.9508 & 0.2557 & -0.1198 \\ 0.0536 & -0.1594 & -0.1071 & 0.9799 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$$
$$+ \begin{pmatrix} -0.0081 & -0.2557 \\ -0.9264 & 0.1071 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$$

[2]The optimal parity space order p_i is the smallest integer that guarantees the existence of the matrix $\Omega_i^{p_i}$.

$$- \rho^1_{c_3}(n) = \begin{pmatrix} 0.2175 & -0.5437 & -0.3625 & 0.7250 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix} + \begin{pmatrix} -0.9425 & 0.3625 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$$

$$- \rho^1_{c_4}(n) = \begin{pmatrix} 0.2175 & -0.5437 & -0.3625 & 0.7250 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix} + \begin{pmatrix} -0.9425 & 0.3625 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$$

A parity-space-based residual bench is implemented to compute on-line the residual vector of the hybrid system. Figures 4.2 and 4.3 show the real-time evolution during 10 seconds of the system residuals when the multimode system is under different modes indicated at the bottom of the figures. Residuals are computed according to the sampling period $T_s = 0.01s$, by the residual bench that takes as input observable variables : the input U and the output Y.
We can verify that residuals of mode q_i are null when the system mode is q_i, $\forall i \in 1..4$. We notice that residuals of modes q_3 and q_4 : $\rho^1_{c_3} = [\tilde{r}_{31}]$ and $\rho^1_{c_4} = [\tilde{r}_{41}]$, are null in mode q_3 as well as in mode q_4. Hence, a diagnosability problem can appear and will be discussed in the following chapter.

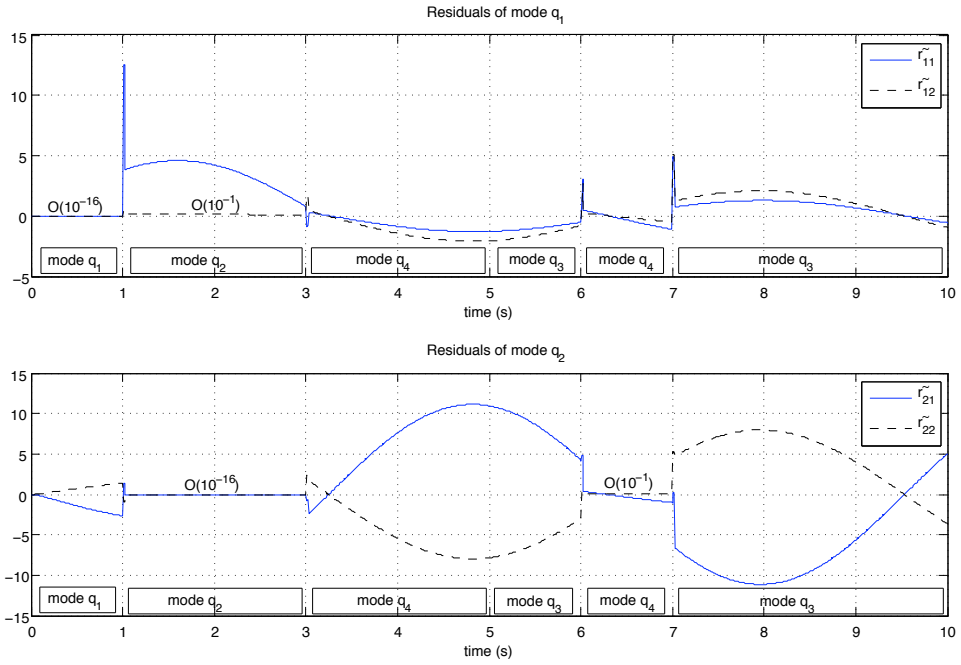

FIG. 4.2 – *Residuals of modes q1 and q2 :* $\rho^1_{c_1} = [\tilde{r}_{11}, \tilde{r}_{12}]^T$ *and* $\rho^1_{c_2} = [\tilde{r}_{21}, \tilde{r}_{22}]^T$

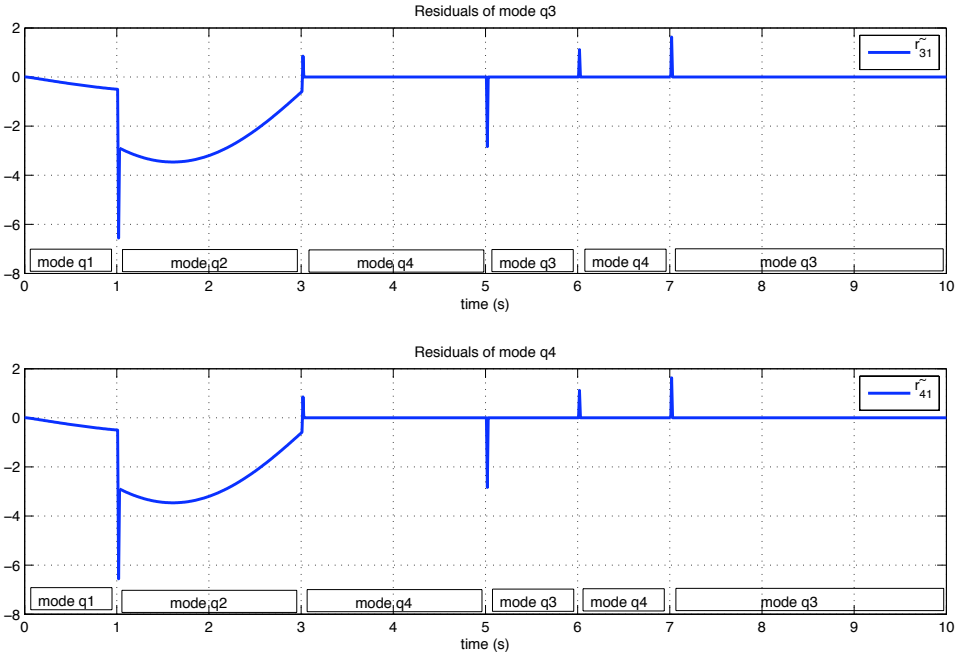

FIG. 4.3 – *Residuals of mode q3 and q4 :* $\rho^1_{c_3} = [\tilde{r}_{31}]$ *and* $\rho^1_{c_4} = [\tilde{r}_{41}]$

4.2.4 Residual filtering

Mode switches are characterized by a spurious jump of the residual values (c.f. Figures 4.2 and 4.3) that is due to the fact that the time-window, over which observations are recorded to evaluate the residuals, overlaps over two modes. Since residuals have been designed for every mode separately, they need at least p_i time steps to settle after a mode change. This may result in a false interpretation of the mode transition that may cause false alarms, in the case when a nominal mode transition is interpreted as a faulty mode transition. This is solved by implementing a residual filter that takes as input residual values computed at every time step, and generating as output clean Boolean residuals that reflect the consistency between model and observed behavior. The filter principle is explained below.

The consistency check is made by comparing the computational and the evaluation forms of the residuals. Given a mode $q_i \in Q$ and $\rho^{p_i}_c(n) = [\tilde{r}_{i1}, ..., \tilde{r}_{iNr(q_i)}]$ the computational form of the residual. Two cases are distinguished :

Noise-free hypothesis

In a noise-free environment ($\rho^{p_i}_{e_i} = 0$), a threshold vector is defined as $\alpha_i = [\alpha_{i1}, ..., \alpha_{iNr(q_i)}]$. The threshold values take into account the computation precision and the relative order of magnitude of the physical variables.

$$r_{ij} = \begin{cases} 0 & \text{if } \tilde{r}_{ij} \leq \alpha_{ij} \\ 1 & \text{otherwise} \end{cases} \tag{4.6}$$

White-Gaussian-Noise hypothesis

In the classical case of a white noise with a normal distribution of the probability density function :
$\epsilon(n) \sim N(0, \sigma^2)$, hence $\epsilon(n, n-p_i) \sim N(0, diag_{p_i+1}(\sigma^2))$ (σ^2 denotes the variance and $diag_{p_i+1}(\sigma^2)$ denotes the diagonal matrix of dimension p_i+1 in which the diagonal values are equal to σ^2).
Consequently the probability density function of the evaluation form has a normal distribution :
$\rho_{e_i}^{p_i}(n) \sim N(0, \Omega_i^{p_i} L^{p_i}(A_i, E_{x_i}, C_i, E_{y_i}) diag(\sigma^2)(L^{p_i}(A_i, E_{x_i}, C_i, E_{y_i}))^T(\Omega_i^{p_i})^T)$

$$r_{ij} = \begin{cases} 0 & \text{if } \tilde{r}_{ij} \sim \rho_{e_{ij}} \\ 1 & \text{otherwise} \end{cases} \tag{4.7}$$

$\rho_{e_{ij}}^{p_i}$ denotes the j^{th} element of $\rho_{e_i}^{p_i}$.

The filter principle

The principle of the residual filter is to hold-on to the current value as long as the Boolean-residual is not computed to a different value during a number of steps specified by a prefixed time-window $T_{Filter} \geq \max_{i=1..m}(p_i)$. The value of T_{Filter} determines the filter sensitivity with respect to the rough residual changes. It is set according to the physical properties of the dynamic system (time response, etc ...) and must be higher than the parity space order of every mode.
Figures 4.4 and 4.5 provide the Boolean residuals for our illustrative example. The threshold has been taken as 10^{-12}.

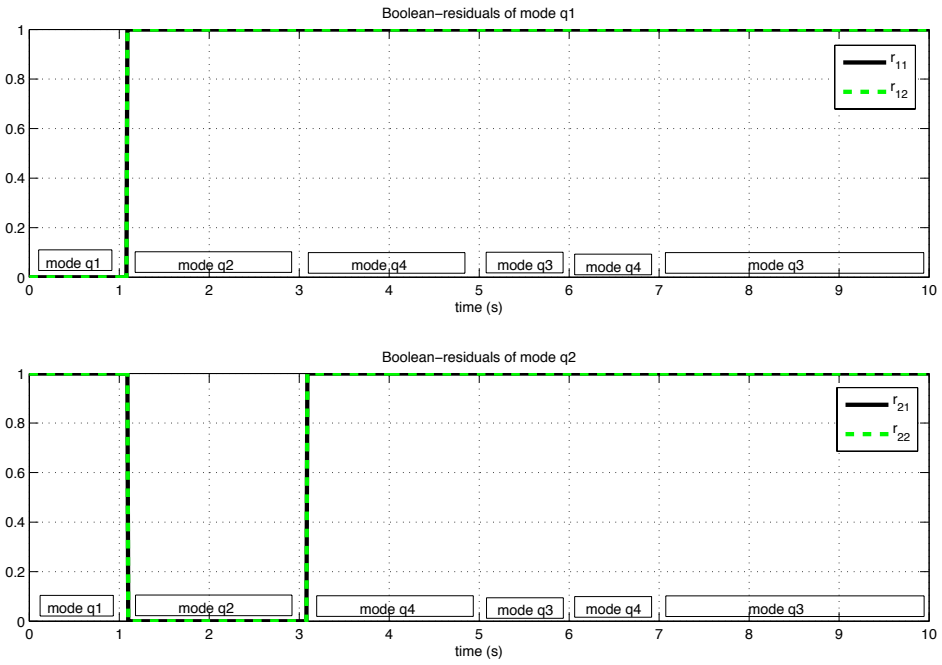

FIG. 4.4 – *Boolean-residuals of modes q1 and q2 (graphs for r_{11} (r_{21}) and r_{12} (r_{22}) are superposed)*

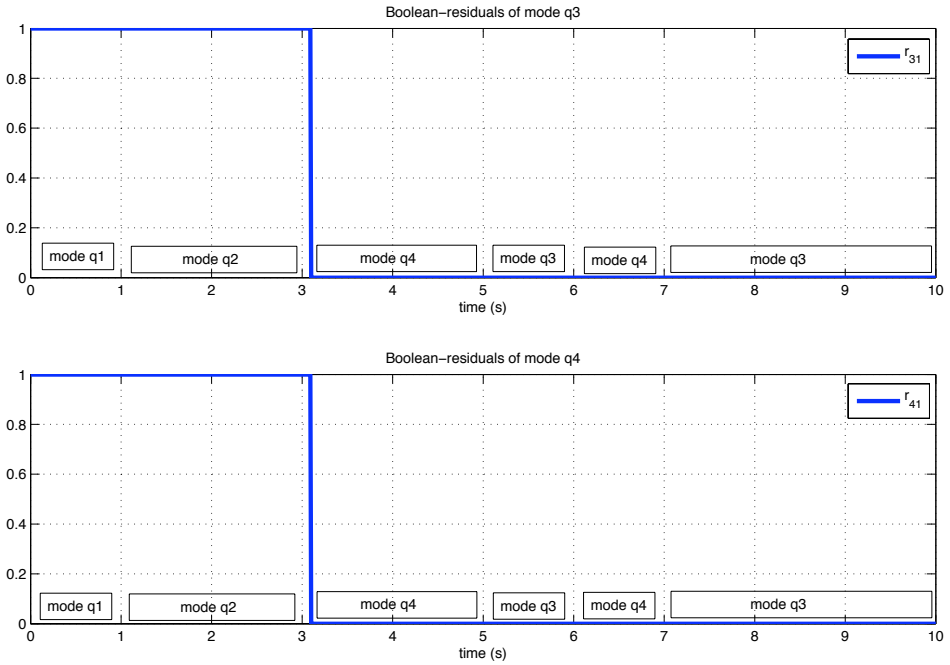

FIG. 4.5 – *Boolean-residuals of modes q3 and q4*

4.2.5 Mirror, reflexive and mode signatures

The concept of fault signature classically defined for continuous systems is now extended to multimode systems. Every operating mode is characterized by a theoretical signature that captures the expected behavior of residuals in this mode. New concepts of mirror and reflexive signatures are defined and lead to the definition of mode signature introduced in Bayoudh *et al.* (2008a).
The q_j-mirror signature of mode q_i is the vector of Boolean-residuals of mode q_j evaluated when the system is in mode q_i. We use the term *mirror* because it represents the signature of q_i seen in mode q_j.

Definition 4.1 *Mirror Signature*
Given the tuple $R^{q_j} = [r_{j1}, r_{j2}, ..., r_{jN_{r(q_j)}}]$ *of the Boolean-residuals associated to mode* q_j*, the* q_j*-mirror signature of mode* q_i *is given by the vector* $S_{i/j} = [s_{1_{i/j}}, ..., s_{N_{r(q_j)_{i/j}}}]^T = [R^{q_j}(\zeta_{OBS_{q_i}})]^T$*, where* $\zeta_{OBS_{q_i}}$ *denotes the incoming observable variables in mode* q_i *i.e. observations that are consistent with the model of mode* q_i*.*

The reflexive signature is a particular case of the mirror signature $S_{i/j}$, with $i = j$.

Definition 4.2 *Reflexive Signature*
The reflexive signature of mode q_i*,* $S_{i/i} = [R^{q_i}(\zeta_{OBS_{q_i}})]^T = [0, 0, ..., 0]^T_{Nr(q_i)}$*, is the vector of Boolean-residuals of mode* q_i*, evaluated with incoming observable variables in mode* q_i*.*

The new concept of mode signature that characterizes a mode is now introduced.

Definition 4.3 *Mode Signature*
The signature of a mode q_i is the vector obtained by the concatenation of all the mirror signatures of q_i, $Sig(q_i) = [S^T_{i/1}, S^T_{i/2}, ..., S^T_{i/i}, ..., S^T_{i/m}]^T$, where m is the number of system modes [3].

In our diagnosis scheme, the real-time mode signature of the system is computed by on-line evaluating the Boolean-residuals of each system mode. The diagnosis of the multimode system is then achieved by comparing the real-time mode signature and the pre-computed theoretical mode signatures.

Illustrative example–mode signatures

The theoretical pre-computed mode signatures of the example of Figure 4.1 are given in Table 4.1. These signatures are determined using the model of each mode to determine the expected behavior of associated residuals.
$S_{1/1}$, $S_{2/2}$, $S_{3/3}$ and $S_{4/4}$ denote the reflexive signatures of modes q_1, q_2,

$Sig(q_1) = \begin{pmatrix} S_{1/1} \\ S_{1/2} \\ S_{1/3} \\ S_{1/4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ - \\ 1 \\ 1 \\ - \\ 1 \\ - \\ 1 \end{pmatrix}$	$Sig(q_2) = \begin{pmatrix} S_{2/1} \\ S_{2/2} \\ S_{2/3} \\ S_{2/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 0 \\ 0 \\ - \\ 1 \\ - \\ 1 \end{pmatrix}$
$Sig(q_3) = \begin{pmatrix} S_{3/1} \\ S_{3/2} \\ S_{3/3} \\ S_{3/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 1 \\ 1 \\ - \\ 0 \\ - \\ 0 \end{pmatrix}$	$Sig(q_4) = \begin{pmatrix} S_{4/1} \\ S_{4/2} \\ S_{4/3} \\ S_{4/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 1 \\ 1 \\ - \\ 0 \\ - \\ 0 \end{pmatrix}$

TAB. 4.1 – *Mode signatures of the underlying continuous system* Ξ

q_3 and q_4 respectively.
$S_{1/2}$, $S_{1/3}$ and $S_{1/4}$ denote the q_2, q_3 and q_4 mirror signatures of mode q_1 respectively and so forth for modes q_2, q_3 and q_4.
In following chapter, these new signature concepts are used to characterize the diagnosability of multimode systems.

4.2.6 Abstraction of the continuous dynamics in terms of discrete events

In our approach, the diagnosis of the hybrid system is performed by the diagnoser built from the underlying discrete-event system enriched with events that capture continuous diagnosis knowledge. To generate these events we propose to use pre-computed theoretical signatures in order to define discrete events associated to each mode signature change. To do this, we define an abstraction function from the continuous domain to the discrete-event domain.

[3] In our approach, nominal and fault modes have the same status and the signature of a given mode anticipates how it should be seen in terms of the indicator tuples of the different modes of the system (including itself).

Assumption 4.1 *We assume that the dynamics of the discrete control inputs are slower than the dynamics of residual generators to guarantee that mode signatures have time to establish between two consecutive discrete events.*

The abstraction function is denoted f_{CS_DES} and defined as follows : for each discrete transition of the underlying discrete-event system, f_{CS_DES} associates an event which represents the change of mode signature.
This function aims to define Σ^{Sig}, as the set of discrete events issued from an abstraction of the continuous dynamics of the multimode system.

$$f_{CS\_DES} : Q \times T(Q,\Sigma) \longrightarrow \Sigma^{Sig}$$

$$(q_i, q_j) \longmapsto \begin{cases} Ro_{ij} \in \Sigma_o^{Sig} \text{ if } Sig(q_i) \neq Sig(q_j) \\ Ruo_{ij} \in \Sigma_{uo}^{Sig} \text{ if } Sig(q_i) = Sig(q_j) \end{cases}$$

- Σ_o^{Sig} is a set of observable events, generated when the mode signature of the source mode is different from the mode signature of the destination mode.
- Σ_{uo}^{Sig} is a set of unobservable events generated when the mode signature of the source mode is equal to the mode signature of the destination mode.
- We define $\Sigma^{Sig} = \Sigma_o^{Sig} \cup \Sigma_{uo}^{Sig}$.

Hybrid language and hybrid trajectories

The abstraction of the continuous dynamics changes in terms of discrete events allows us to define the language of the hybrid system, which describes the evolution of the system behavior.
We denote by $\Sigma_{hyb} = \Sigma \cup \Sigma^{Sig}$ the alphabet that contains "natural" discrete events and events modeling the signature switches.
Σ_{hyb} can be partitioned into $\Sigma_{hyb} = \Sigma_{hyb_o} \cup \Sigma_{hyb_{uo}}$ with $\Sigma_{hyb_o} = \Sigma_o \cup \Sigma_o^{Sig}$ and $\Sigma_{hyb_{uo}} = \Sigma_{uo} \cup \Sigma_{uo}^{Sig}$.
The behavior of the hybrid system is modeled by the prefix-closed language $L(S) \subseteq \Sigma_{hyb}^*$ over the event alphabet Σ_{hyb}, where Σ_{hyb}^* denotes the set of all finite strings of elements of the set Σ_{hyb} including the empty string (Σ_{hyb}^* is called the Kleene-Closure of Σ_{hyb} as presented in Ramadge and Wonham (1989)). A trajectory of the hybrid system is represented by a string of events of the hybrid alphabet Σ_{hyb}.

The behavior automaton

The hybrid language $L(S)$ can be represented by its finite state generator representation (c.f. Ramadge and Wonham (1989)) called the *behavior automaton* denoted $B_A(S) = (Q_{beh}, \Sigma_{hyb}, T_{beh}, q_0)$ which generates both "natural" discrete events and events issued from the abstraction of continuous dynamics (signature changes).
The behavior automaton construction is achieved as follows : let Q_t denote the set of *transient* modes that model the continuous dynamic reaction after the occurrence of a discrete event. We define the bijective function f_t that associates a transient mode to each mode change represented as a

pair of modes (source and destination modes). The set of transient modes is obtained as follows :

$$f_t : Q \times T(Q,\Sigma) \longrightarrow Q_t$$
$$(q_i, q_j) \longmapsto q_{ij}$$

The set of modes of the behavior automaton is $Q_{beh} = Q \cup Q_t$ and the partial transition function T_{beh} is defined as follows :

$$T_{beh} \subseteq (Q_{beh} \times \Sigma_{hyb} \longrightarrow Q_{beh})$$

$$(q,\sigma) \longmapsto \begin{cases} f_t(q, T(q,\sigma)) \text{ if } q \in Q \text{ and } \sigma \in \Sigma \\ (f_t^{-1})_2(q) \text{ if } q \in Q_t \text{ and } \sigma \in \Sigma^{Sig} \end{cases}$$

$(f_t^{-1})_2$ denotes the second component of the inverse function of f_t.
In practice, non observable signature switches (Σ_{uo}^{Sig}) are useless because they do not convey additional information, hence they are not considered as well as corresponding transient modes (c.f. Figure 4.6 in which Ruo_{34} and Ruo_{43} do not appear neither q_{34} and q_{43}). In this case, Σ_{hyb} is defined as $\Sigma \cup \Sigma_o^{Sig}$ (i.e. $\Sigma_{hyb_{uo}} = \Sigma_{uo}$).
For illustration, Figure 4.6 provides the behavior automaton of the hybrid system shown in Figure 4.1. q_{13}, q_{12}, q_{32} and q_{24} are transient modes added to the mode automaton to model the response of the underlying continuous system to event occurrences.

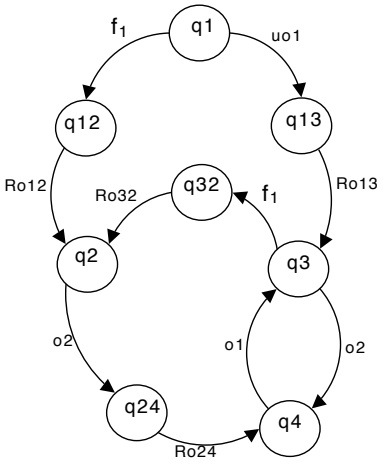

FIG. 4.6 – *The behavior automaton of the hybrid system S*

4.2.7 Extension of the diagnoser approach to hybrid systems diagnosis

The diagnosis of the hybrid system is achieved by extending the diagnoser approach to hybrid systems. The diagnoser of the hybrid system is a finite state machine built from the behavior automaton as follows :
First, we define a set of fault labels $\Delta_f = \{F_1, F_2, ..., F_m\}$, where m is the number of different fault types in the system. The set of possible fault labels is defined as $\Delta = 2^{\Delta_f}$.
Notice that the empty-set label $\emptyset \in \Delta$ should be interpreted as representing the normal behavior of the system. A label of the form $\{F_i, F_j\}$ should

be interpreted to mean that at least one fault of type i and at least one fault of type j has occurred.
Given $s \in \Sigma^*_{hyb}$ a string of events, "$\Sigma_{F_i} \in s$" should be interpreted to mean that at least one fault event of type i belongs to s.
Let s_f denote the final event of a string s and $L(S,q)$ the set of all strings that originate from state $q \in Q_{beh}$.
We define : $L_o(S,q) = \{s \in L(S,q) \mid s = u\sigma, u \in \Sigma^*_{hyb_{uo}}, \sigma \in \Sigma_{hyb_o}\}$
and $L_\sigma(S,q) = \{s \in L_o(S,q) \mid s_f = \sigma\}$.
$L_o(S,q)$ denotes the set of all strings that originate from the state q and end at the first observable event.
$L_\sigma(S,q)$ denotes those strings in $L_o(S,q)$ that end at the particular observable event σ.
$Q_{beh_o} = \{q_0\} \cup \{q \in Q_{beh}, \exists(q',\sigma) \in Q_{beh} \times \Sigma_{hyb_o} \mid T_{beh}(q',\sigma) = q\}$ denotes the set of observable states.
We define the label propagation function :
$LP : Q_{beh_o} \times \Delta \times \Sigma^*_{hyb} \rightarrow \Delta$ as :

$$LP(q,l,s) = \begin{cases} \emptyset \text{ if } l = \emptyset \text{ and } \forall i, \Sigma_{F_i} \notin s \\ \{F_i | F_i \in l\} \cup \{F_i | \Sigma_{F_i} \in s\} \text{ otherwise} \end{cases}$$

The diagnoser of the hybrid system is a deterministic finite state machine built from the behavior automaton, $Diag(B_A(S)) = (Q_D, \Sigma_D, T_D, q_{D_0})$ with :

- $q_{D_0} = \{(q_0, \emptyset)\}$ is the initial state of the diagnoser (we assume that the system S is normal to start with).
- $\Sigma_D = \Sigma_{hyb_o}$ is the set of all observable events of the system.
- $Q_D \subseteq 2^{Q_{beh_o} \times \Delta}$ is the set of states of the diagnoser (states reachable from q_{D_0} under T_D). The states of the diagnoser provide the set of diagnosis candidates as a set of couples whose first element refers to the state of the behavior automaton and the second is a label providing the set of faults on the path leading to this state. In other words, an element $q_D \in Q_D$ is a set of the form $q_D = \{(q_1,l_1),(q_2,l_2),...,(q_n,l_n)\}$, where $q_i \in Q_{beh_o}$ and $l_i \in \Delta$.
- $T_D \subseteq Q_D \times \Sigma_{hyb_o} \rightarrow Q_D$ is the partial transition function of the diagnoser defined as follows :
  $T_D(q_D,\sigma) = \bigcup_{\substack{(q,l) \in q_D \\ s \in L_\sigma(S,q)}} \{(T_{beh}(q,s), LP(q,l,s))\}$

The diagnoser of the hybrid system (c.f. Figure 4.1) is provided in Figure 4.8. Notice that the notion of fault label can be extended to other non unobservable events if the historic of unobservable event occurrence is needed.

4.2.8 The hybrid diagnosis scheme based on the diagnoser of the hybrid system

Figure 4.7 provides our hybrid diagnosis scheme implemented with MATLAB/SIMULINK :

- the block *HYBRID SYSTEM* corresponds to the system model. Underlying continuous behaviors are modeled by means of state-space

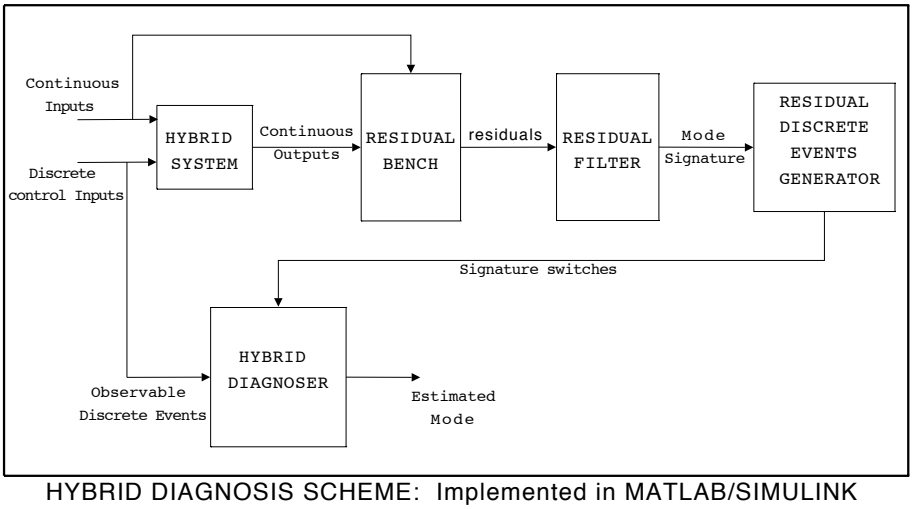

FIG. 4.7 – *Diagnosis scheme by coupling discrete-event and continuous techniques*

models and discrete-event dynamics are modeled as a finite state machine by means of an incidence-matrix representation.

– the block *RESIDUAL BENCH* computes the vector of residuals associated to each mode. As output, it gives the on-line evolution of the residuals.
– the block *RESIDUAL FILTER* filters the system residuals to obtain the Boolean-residuals. By putting together these Boolean-residuals, we obtain the real-time mode signature of the hybrid system.

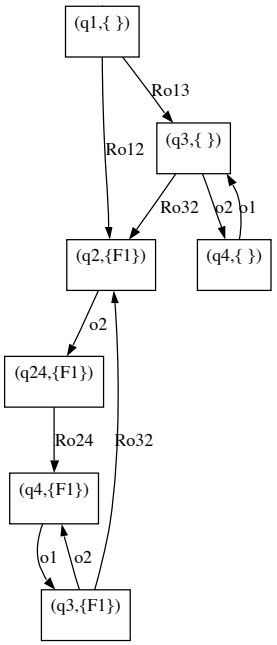

FIG. 4.8 – *The diagnoser of the hybrid system built from the behavior automaton*

– the block *RESIDUAL DISCRETE-EVENT GENERATOR* generates an observable discrete event (from Σ_o^{Sig}) when the real-time mode signature changes. The generator is modeled as a finite state machine,

each state corresponds to a theoretical mode signature, the generated discrete events are linked with state transitions.
– the block *HYBRID DIAGNOSER* performs the on-line mode tracking of the hybrid system. It takes as input observable "natural" discrete events, and observable events that capture the continuous diagnosis knowledge. The hybrid diagnoser is modeled as a finite state machine built off-line from the behavior automaton by applying the diagnoser approach [4].

4.2.9 Illustrative example : mode tracking of the hybrid system

Let us consider again the illustrative example shown in Figure 4.1. The hybrid system is implemented in a MATALAB/SIMULINK block and hybrid diagnosis is achieved as explained in Figure 4.7. The diagnoser of the hybrid system is shown in Figure 4.8.
The system mode is tracked at every time step during the simulation time given by $T_{simulation} = 10s$ and the sampling period $T_s = 0.01s$. The filter sensitivity is set as $T_{Filter} = 0.07s$. Two different scenarios are tested :

Scenario 1

The system starts from the initial mode q_1 and follows the discrete trajectory : $[(f_1, t = 1s), (o_2, t = 3s), (o_1, t = 5s), (o_2, t = 6s), (o_1, t = 7s)]$.
The evolution of the residuals and the Boolean-residuals of modes q_1, q_2, q_3 and q_4 have been already given in Figures 4.2, 4.3, 4.4 and 4.5 respectively. On-line mode tracking is performed thanks to the hybrid diagnoser. The

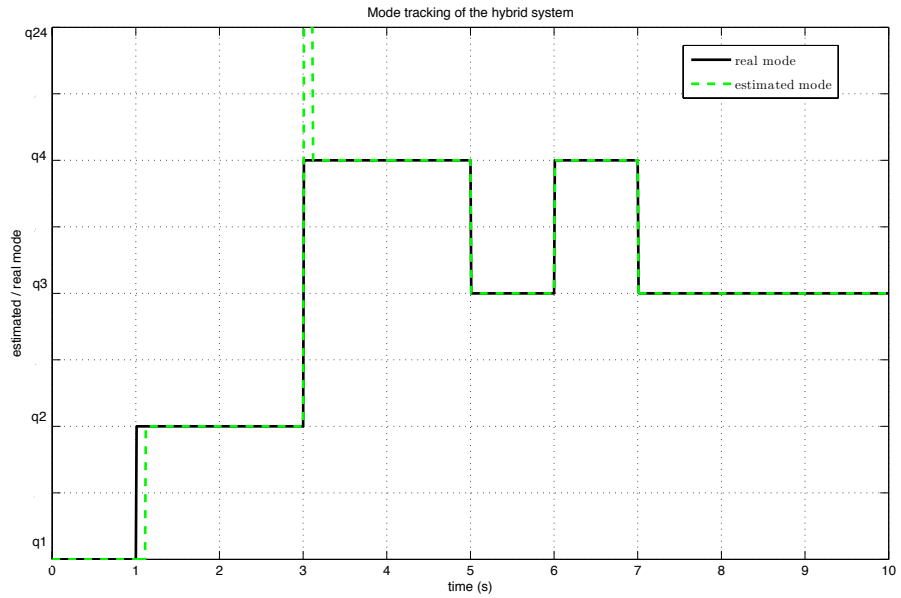

FIG. 4.9 – *Scenario 1 : mode tracking of the hybrid system*

estimated mode as well as the real mode are shown in Figure 4.9. Table

[4]We use the *DIADES* software from Pencolé (2006) to build the diagnoser of the hybrid system from the behavior automaton.

4.2 shows the mode transition detection times corresponding to event occurrences. We notice that the occurrence of the fault event f_1 is detected after a delay equal to $0.13s$.

Event occurrence	Mode transition time
$(f_1, t = 1s)$	$q_1 \xrightarrow{t=1.13s} q_2$
$(o_2, t = 3s)$	$q_2 \xrightarrow{t=3.02s} q_{24} \xrightarrow{t=3.13s} q_4$
$(o_1, t = 5s)$	$q_4 \xrightarrow{t=5.02s} q_3$
$(o_2, t = 6s)$	$q_3 \xrightarrow{t=6.02s} q_4$
$(o_1, t = 7s)$	$q_4 \xrightarrow{t=7.02s} q_3$

TAB. 4.2 – *Mode tracking scenario 1 : mode transition detection times*

Scenario 2

The system starts from the initial mode q_1 and follows the discrete trajectory : $[(uo_1, t = 3s), (f_1, t = 5s), (o_2, t = 7s), (o_1, t = 9s)]$. The estimated mode as well as the real mode are shown in Figure 4.10. Table 4.3 shows mode transition detection times corresponding to event occurrences. We

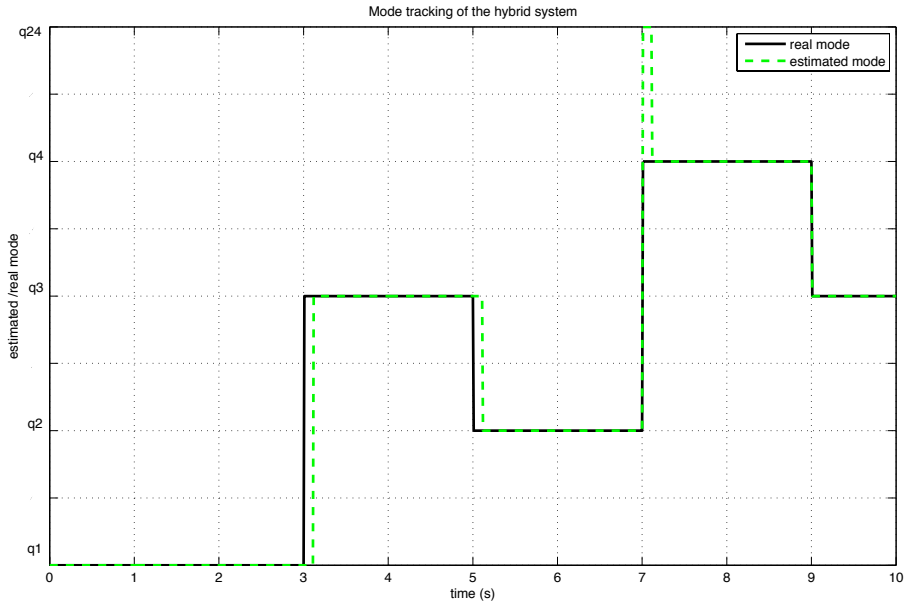

FIG. 4.10 – *Scenario 2 : mode tracking the hybrid system*

notice that the occurrence of the fault event f_1 and the unobservable event uo_1 are detected after a delay equal to $0.13s$.

Let us notice that the hybrid diagnoser is able to follow the system mode, even after unobservable event occurrence (f_1 and uo_1). The hybrid diagnoser can also diagnose modes q_3 and q_4 that have the same mode signature. This is achieved by coupling continuous and discrete informations. Notice that the diagnoser tracks the system mode with a delay. Two types of delay are distinguished :

Event occurrence	Mode transition time
$(uo_1, t = 3s)$	$q_1 \xrightarrow{t=3.13s} q_3$
$(f_1, t = 5s)$	$q_3 \xrightarrow{t=5.13s} q_2$
$(o_2, t = 7s)$	$q_2 \xrightarrow{t=7.02s} q_{24} \xrightarrow{t=7.13s} q_4$
$(o_1, t = 9s)$	$q_4 \xrightarrow{t=9.02s} q_3$

Tab. 4.3 – *Mode tracking scenario 2 : mode transition detection times*

- a small delay ($2.T_s$) due to the computation time, it represents the time needed to upload the observed signature.
- a delay due to the sensitivity of the residual filter T_{Filter} that represents the time needed by the algorithm to filter the system residuals. This delay is linearly dependent of the sampling period.

The second type of delay is avoided if the occurred event is observable, indeed in this case the discrete-event knowledge is sufficient to detect mode transition. However, after the occurrence of an unobservable event, the delay is due to the two types of delays ($T_{Filter} + 3.(2.T_s)$), hence, the delay can be reduced if we improve the computation resolution.
In these two scenarios, the system mode returned by the hybrid diagnoser at every sampling time is unique. It means that the diagnosis is precise at every sampling time of the simulation, thus active diagnosis is not needed.

4.3 Geometrical Interpretation in the space of system modes

In this section, we propose a geometrical representation of hybrid systems, in the vectorial space $\mathbb{R}^m$, where m is the number of operational system modes as proposed in Bayoudh *et al.* (2007)[5].
The vectorial space $\mathbb{R}^m$ is described by the vectorial base $B = (\vec{q}_1, \vec{q}_2, ..., \vec{q}_m)$ and it is called *the space of modes*, with $\vec{q}_1 = [1, 0, 0, ..., 0]_m$, $\vec{q}_2 = [0, 1, 0, ..., 0]_m$, ..., $\vec{q}_m = [0, 0, 0, ..., 1]_m$.
Given $R^{q_i} = [r_{i1}, ..., r_{iNr(q_i)}]^T$ the vector of Boolean-residuals of mode $q_i \in Q$. Let $||R^{q_i}|| = \sqrt{(\sum_{j=1..N_{r(q_i)}} r_{ij}^2)}$ denote the Euclidian norm.
When the system mode is q_i, the associated vector of Boolean-residuals is zero, i.e. $R^{q_i} = [0, 0, ..., 0]^T$. However the reciprocal is not true, it depends on the diagnosability property of the system as will be discussed in Chapter 5.
We propose to describe the behavior of the hybrid system S in the space of modes at a sampling time n by the linear mapping F defined as follows :

$$F(S)(n) = ||R^{q_1}||(n)\vec{q}_1 + ||R^{q_2}||(n)\vec{q}_2 + ... + ||R^{q_m}||(n)\vec{q}_m \qquad (4.8)$$

When the system evolves in the mode q_i : $||R^{q_i}|| = 0$, then :

$$F(S)(n) = ||R^{q_1}||(n)\vec{q}_1 + ||R^{q_{i-1}}||(n)\vec{q}_{i-1} + ||R^{q_{i+1}}||(n)\vec{q}_{i+1} + ... + ||R^{q_m}||(n)\vec{q}_m$$

[5]It will be shown in Chapter 5 that this vectorial representation in the space of modes offers a framework to geometrically interpret the diagnosability of multimode systems.

Consequently the system evolves in the space region denoted $\mathcal{R}_{eg}^{q_i}$ included in the subspace of dimension $m-1$ orthogonal to the vector $\vec{q}_i$ denoted : $\vec{q}_i^{\perp}$, hence, $\mathcal{R}_{eg}^{q_i} \subseteq \vec{q}_i^{\perp}$.

The dimension of $\mathcal{R}_{eg}^{q_i}$ depends on the diagnosability properties of the

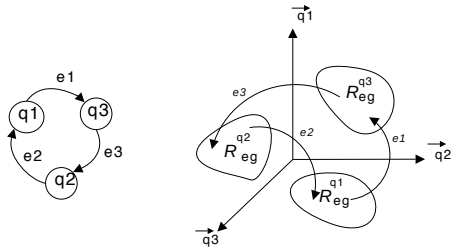

Fig. 4.11 – *Example of a 3-dimensional space of modes*

multimode system. It will be discussed in Chapter 5. For illustration, Figure 4.11 shows an example of the 3-dimensional space of modes.

4.4 Hybrid state estimation through synergic mode-set focusing

The work presented above in this chapter proposes a passive diagnosis approach in which the diagnosis problem is formulated as a mode estimation problem solved through a hybrid parity-based method. However it does not provide the continuous state estimation, which may be a problem in case an appropriate feedback reaction is needed on the system (for active diagnosis or control purposes). On the other hand, hybrid estimation schemas that account both for the continuously-valued state evolution and the interleaved discrete mode changes suffer from a blow up of the number of estimation hypothesis to be considered.

This section summarizes a piece of work arising from a collaboration with Graz University of Technology, namely Prof. M. Hofbaur and his PhD student T. Rienmüller. The work couples our diagnosis approach to the filtering approach of Hofbaur and Williams (2004). It results in a novel scheme that uses the hybrid parity-based method as a mode focusing procedure and then applies hybrid estimation on the resulting reduced number of hypotheses. The advantages of the mixed method are on both sides : it boosts the mode identification time and the convergence of continuous state estimation. This work has been accepted for presentation in the IFAC Safeprocess 09 Symposium (c.f. Rienmüller *et al.* (2009)). The principle of our mixed method is to perform continuous estimation in two ways, even-though this may appear as redundant. The parity-space based diagnosis technique is used as an abstraction operator of the continuous evolution and is used to supply a focused mode estimation through a discrete-event diagnoser (cf. section 4.2.6). This restricted set of possible mode hypotheses is given as input to the additional continuous estimation scheme that supplies the neglected continuous state estimate through a traditional filtering based hybrid estimation technique. This is illustrated in Figure 4.12, which can be compared to Figure 4.7.

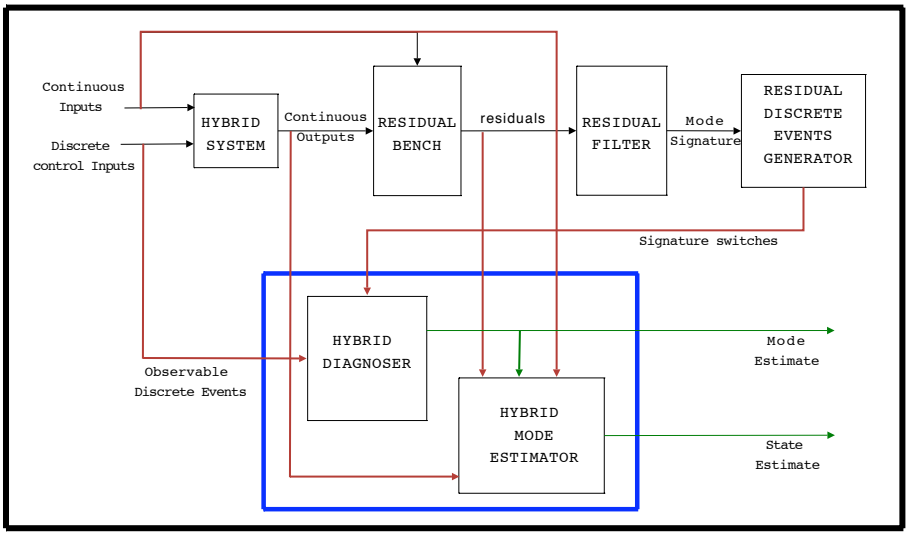

FIG. 4.12 – *Mixed method architecture*

4.4.1 Hybrid Estimation

Given a hybrid model S, the discrete-time sequence of noisy (continuous) observations $\{Y(1), \ldots, Y(n)\}$, the sequence of observable events $\{\sigma_1, \ldots, \sigma_n\}$ and the actuated control inputs $\{U(1), \ldots, U(n)\}$, the hybrid estimation problem estimates the mode hybrid state that is composed of the mode of operation $q_i \in Q$ and the continuous state $X_i(n)$ for time-step n. Hybrid estimation must be performed under partial observations and account for model uncertainties, for instance the mode evolution of the automaton cannot generally be fully observed. As a consequence, a hybrid estimator has to consider all possible mode sequences with its associated continuous evolution that are consistent with the actuation and observations. This results in an inevitable blow up of the number of hypotheses, which has led the community to propose different suboptimal hybrid estimation schemes. Early solutions to the hybrid estimation problem such as the multi-model IMM algorithm (c.f. Blom and Bar-Shalom (1998)) track hypotheses over a limited number of time-steps only and merge the continuous estimates according to a measure of likelihood. This likelihood is mostly drawn from the continuous filters and expresses the level of agreement between the estimate and the observations but might also include prior transition probability information, if available. Several other strategies have been proposed later, ranging from hierarchical approaches (c.f. Verma *et al.* (2003)) to mixed sampling and search (c.f. Blackmore *et al.* (2005)). The hybrid estimation algorithm (HME) (c.f. Hofbaur and Williams (2004)) that we consider for this work, uses the likelihood measure to *focus* on the set of most likely hypotheses. The first best hypotheses are obtained thanks to a focused search strategy, leading to an any-time any-space algorithm. Although the method was shown to operate successfully on systems with a quite large number of modes, its efficiency can be significantly improved by focusing on possible hypothesis.

4.4.2 Hybrid Estimation through mode set focusing

The mode estimate provided by our hybrid parity-based method (cf. Section 4.2.6) is now coupled with the associated continuous state estimate provided by HME (cf. Section 4.4.1). It is interesting to notice that when the mode estimator output a single mode estimate for the current mode of operation, the continuous state estimation problem reduces to a standard filtering problem. However, it is generally the case that, due to uncertainties and scare observations, mode estimation takes the form of a set of possible modes.
In the following, it is shown that this coupling is valuable in all aspects, as it synergically contributes both to accelerate mode change detection and to improve the continuous state estimation quality. The first improvement deals with the delay required by the mode estimator to perform mode identification. Indeed, although mode change detection can be drawn from abrupt residual changes almost instantly, mode identification is delayed by two factors :

- the p time-steps required by the observation window to report single-mode observed data
- the additional T_{filter} time-steps required by the algorithm to filter the system residuals.

This delay would of course introduce an error on the continuous state estimate as well, resulting in a low quality overall procedure. To avoid the above delay, the idea is to not wait until the mode estimator settles upon a new or set of new modes of operation, but to generate the hypotheses to be followed by HME from the previous mode estimate. As a result, we get the following nice features :

- continuous state estimates for all hypotheses under consideration with their associated likelihood values are immediately obtained.
- the adaptation delay of the continuous estimate is avoided by tracking multiple hypotheses and the hybrid estimator immediately provides the correct continuous estimate.

Of course, an analogous interaction can be used whenever the mode estimator settles upon a focused set of modes. This additional evidence can easily be included in the multi-mode estimation scheme as additional focusing method that contributes to the estimation quality but also reduces the computational effort that is necessary for hybrid estimation.

Chapter 5

Framework for diagnosability Analysis of Hybrid Systems

Contents

The "passive" diagnosis approach introduced in Chapter 4 does not take into account the diagnosability properties of the hybrid system. Consequently, the uniqueness of the diagnosis is not guaranteed. Indeed, diagnosability is the property that ensures that the system state can be precisely diagnosed after the occurrence of a fault. In an autonomy context, in particular for satellites, the diagnosability properties can be used to guide the active diagnosis process in order to disambiguate an ambiguous estimated situation. The diagnosability definition depends mainly on the system model, the diagnosis approach and the observation system (the manner in which the system is observed). Diagnosability of discrete-event and continuous systems as well as some work directions for hybrid systems have been recalled in Chapter 3.
In this chapter, we propose a formulation of the diagnosability of multimode systems on one hand, and of hybrid systems on the other hand. This formulation is based on the concepts of mirror, reflexive and mode signatures introduced in Chapter 4.

5.1 Diagnosability of multimode systems

5.1.1 From mode signatures to multimode system diagnosability characterization

The concept of mode signature presented previously in Chapter 4 leads us to the characterization of the diagnosability of multimode systems.
Let us notice that in our approach, the faulty behaviors are modeled by fault modes. A given fault, in the classical sense, corresponds to a set of fault modes in which this fault is present. In this thesis, diagnosability (of multimode systems) is analysed at the level of fault modes, which is somehow more precise than at the level of faults. Indeed, whereas the signature of a mode is reduced to one single tuple, the signature of a fault is in general a set of tuples.

Example 5.1 *Let consider for instance a system composed by a valve and a pump that has two operating modes, pump "on" and pump "off". Then, the fault "f_ValveBlocked" has two corresponding fault modes "Mode_ValveBlocked_PumpOn" and "Mode_ValveBlocked_PumpOff" (c.f. Figure 5.1).*
Moreover, notice that we have $Sig(f\_ValveBlocked) = \{Sig(Mode\_ValveBlocked\_PumpOn), Sig(Mode\_ValveBlocked\_PumpOff)\}$.

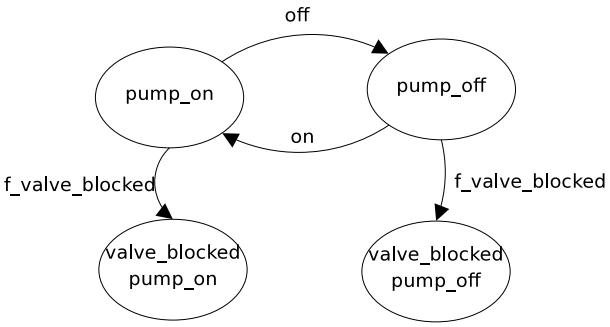

Fig. 5.1 – *The relation between fault signature and mode signature*

By analogy with fault diagnosability of continuous systems, the concepts of mode and fault diagnosability of multimode systems are defined as follows :

Definition 5.1 *Two modes* q_i *and* q_j $(i \neq j)$ *are diagnosable if* $Sig(q_i) \neq Sig(q_j)$.
The multimode system Ξ *is diagnosable if and only if all pairs of modes* q_i *and* q_j, $i \neq j$, *are diagnosable.*

Definition 5.2 *The signature of a fault* F *is defined as the set of the signatures of all the system modes in which the fault is present.*

Definition 5.3 *In the case in which the model does not account for reparability actions, i.e. actions that repair the faults, the signature of a fault* F_i *is equal to the set of the signatures of all possible destination modes after the occurrence of the fault event* f_i. *Formally :*
$Sig(F_i) = \bigcup_{\substack{k \in 1..m \\ u \in \Sigma^*}} \{Sig(T(q_k, f_i u))\}$.

Definition 5.4 *Two faults F_i and $F_j, i \neq j$ are diagnosable if $Sig(F_i) \cap Sig(F_j) = \emptyset$.*

Then, the diagnosability of two modes q_i and q_j of the multimode system is interpreted along two complementary ways through the definitions of *mutual* and 3^{rd} *diagnosability* :

Definition 5.5 *Mutual Diagnosability*
Two modes q_i and q_j, $i \neq j$, are not mutually diagnosable if :
$S_{i/i} = S_{i/j} = [0, 0, ..., 0]^T_{Nr(q_i)}$ *and* $S_{j/j} = S_{j/i} = [0, 0, ..., 0]^T_{Nr(q_j)}$.

Mutual diagnosability is equivalent to *mode discernability* as defined in Cocquempot *et al.* (2004).

Definition 5.6 *3^{rd}-Diagnosability*
Two modes q_i and q_j are q_k-3^{rd}-diagnosable if they have different signatures with respect to the q_k mode , i.e. they have two different q_k-mirror signatures, $k \neq i, j$.
Formally, q_i and q_j, $i \neq j$, are q_k-3^{rd}-mirror diagnosable if $S_{i/k} \neq S_{j/k}$.
Two modes q_i and q_j, $i \neq j$, are 3^{rd}-diagnosable if $\exists k \neq i, j$ such as $S_{i/k} \neq S_{j/k}$.
The multimode system is 3^{rd}-diagnosable if for all pairs of modes q_i and q_j, $i \neq j$, there exists $k_{i,j} \neq i, j$ such that $S_{i/k_{i,j}} \neq S_{j/k_{i,j}}$.

Then, we have the following result :

Theorem 5.1 *Two modes q_i and q_j, $i \neq j$ are diagnosable if and only if they are mutually diagnosable or 3^{rd}-diagnosable.*

Proof. Consider two modes q_i and q_j, $i \neq j$.
Let $Sig(q_i) = [S^T_{i/1}, S^T_{i/2}, ..., S^T_{i/i}, ..., S^T_{i/m}]^T$
and $Sig(q_j) = [S^T_{j/1}, S^T_{j/2}, ..., S^T_{j/j}, ..., S^T_{j/m}]^T$
q_i and q_j are diagnosable if and only if $Sig(q_i) \neq Sig(q_j)$
$\Leftrightarrow \exists k \in 1..m$ such as $S^T_{i/k} \neq S^T_{j/k}$
$\Leftrightarrow q_i$ and q_j are 3^{rd} (if $k \neq i, j$) or mutually (if $k = i$ or $k = j$) diagnosable. □

Consequently, the multimode system is diagnosable if and only if for every pair of modes (q_i, q_j), $i \neq j$ mutual or/and 3^{rd}-diagnosability holds.

Mutual and 3^{rd}-diagnosability : illustrative example

Consider again the example (Figure 4.1) introduced in chapter 4. We focus on the diagnosability of the underlying continuous system. Let us consider the mode signatures provided in Table 4.1. We notice that modes q_1 and q_2 are mutually diagnosable by the fact that $S_{1/1} \neq S_{2/1}$ (or $S_{1/2} \neq S_{2/2}$). Hence they are diagnosable.
We notice that modes q_3 and q_4 are non diagnosable because they have the same mode signature. It can be interpreted as follows :

- $S_{3/3} = S_{4/3}$ and $S_{4/4} = S_{3/4}$, thus q_3 and q_4 are non mutually diagnosable.
- $S_{3/1} = S_{4/1}$ and $S_{3/2} = S_{4/2}$, thus q_3 and q_4 are non 3^{rd}-diagnosable.

Therefore the underlying continuous system is not diagnosable w.r.t the diagnosability definition of multimode systems.

5.1.2 Structural conditions for mutual and 3^{rd} diagnosability in the case of linear continuous behaviors

In the case of multimode systems with linear continuous behaviors in operating modes, structural conditions for mutual can be established. Consider a multimode system modeled in the state space as given by Equation 4.1. We take an identical parity order for all modes ($p = \underset{i\in 1..m}{Max}\{p_i\}$). For sake of simplicity, we consider the case in which there is no noise and no perturbation ($\forall i \in 1..m, \rho^p_{c_i}(n) = 0, n \in \mathbb{N}$). The computational form in mode q_i is given by Equation 4.4 :

$$\rho^p_{c_i}(n) = \Omega^p_i Y^p(n) - \Omega^p_i L^p_i(A_i, B_i, C_i, D_i)U^p(n)$$

If we replace Y^p by the outputs given by the theoretical model of mode q_j (given by Equation 4.3, i.e. $Y^p(n) = O^p_j X_j(n-p) + L^p_j(A_j, B_j, C_j, D_j)U^p(n)$), we obtain :

$$\rho^p_{c_i}(n) = \Omega^p_i O^p_j X_j(n-p) + \Omega^p_i [L^p_j(A_j, B_j, C_j, D_j) - L^p_i(A_i, B_i, C_i, D_i)]U^p(n)$$

Mutual diagnosability

Theorem 5.2 *Two modes q_i and q_j, $i \neq j$ modeled in the state space as given by Equation 4.1 are non mutually diagnosable if and only if : $Ker^1([O^p_j]^T) = Ker([O^p_j]^T)$ and $Ker([O^p_i]^T) \subseteq Ker([L^p_j(A_j, B_j, C_j, D_j) - L^p_i(A_i, B_i, C_i, D_i)]^T)$.*

Proof. Modes q_i and q_j are non mutually diagnosable $S_{i/i} = S_{i/j} = [0, 0, ..., 0]^T_i$ and $S_{j/j} = S_{j/i} = [0, 0, ..., 0]^T_j$ if and only if :
$\forall n \in \mathbb{N}, \rho^p_{c_i}(n) = \rho^p_{c_j}(n) = 0, \forall (U^p, Y^p)$ the incoming inputs/outputs in mode q_i or q_j (this is generalized in Proposition 5.1).
$\Leftrightarrow \forall U^p, \forall X_i(n-p), \forall X_j(n-p)$:

$$\Omega^p_i O^p_j X_j(n-p) + \Omega^p_i [L^p_j(A_j, B_j, C_j, D_j) - L^p_i(A_i, B_i, C_i, D_i)]U^p(n) = 0$$

and

$$\Omega^p_j O^p_i X_i(n-p) + \Omega^p_j [L^p_i(A_i, B_i, C_i, D_i) - L^p_j(A_j, B_j, C_j, D_j)]U^p(n) = 0$$

$\Leftrightarrow$

$$\Omega^p_i O^p_j = \Omega^p_j O^p_i = 0$$

and

$$\Omega^p_i [L^p_j(A_j, B_j, C_j, D_j) - L^p_i(A_i, B_i, C_i, D_i)] = 0$$

Since Ω^p_i (Ω^p_j) is defined as $\Omega^p_i O^p_i = 0$ ($\Omega^p_j O^p_j = 0$), we obtain the following result :
$Ker([O^p_j]^T) = Ker([O^p_j]^T)$ and
$Ker([O^p_i]^T) \subseteq Ker([L^p_j(A_j, B_j, C_j, D_j) - L^p_i(A_i, B_i, C_i, D_i)]^T)$ □

[1] Ker(M) denotes the null space of a given matrix M.

3^{rd}-Diagnosability

Theorem 5.3 *Two modes q_i and q_j, $i \neq j$ modeled in the state space as given by Equation 4.1 are non $q_k - 3^{rd}$-diagnosable if :*
$Ker([O_k^p]^T) \subseteq Ker([O_i^p]^T)$ and $Ker([O_k^p]^T) \subseteq Ker([O_j^p]^T)$ and $Ker([O_k^p]^T) \subseteq Ker([L_j^p(A_j, B_j, C_j, D_j) - L_i^p(A_i, B_i, C_i, D_i)]^T)$.

Proof. q_i and q_j are non $q_k - 3^{rd}$-diagnosable ($S_{i/k} = S_{j/k}$) if :
$\forall n \in \mathbb{N}, \rho_{c_k}^p(n)$ is the same, $\forall (U^p, Y^p)$ the incoming inputs/outputs in mode q_i or q_j.
$\Leftrightarrow \forall U^p, \forall X_i(n-p), \forall X_j(n-p)$:

$$\Omega_k^p O_i^p X_i(n-p) + \Omega_k^p [L_i^p(A_i, B_i, C_i, D_i) - L_k^p(A_k, B_k, C_k, D_k)] U^p(n) =$$

$$\Omega_k^p O_j^p X_j(n-p) + \Omega_k^p [L_j^p(A_j, B_j, C_j, D_j) - L_k^p(A_k, B_k, C_k, D_k)] U^p(n)$$

$\Leftrightarrow \forall U^p, \forall X_i(n-p), \forall X_j(n-p)$:

$$\Omega_k^p O_i^p X_i(n-p) + \Omega_k^p [L_i^p(A_i, B_i, C_i, D_i) - L_k^p(A_k, B_k, C_k, D_k)] U^p(n) -$$

$$\Omega_k^p O_j^p X_j(n-p) + \Omega_k^p [L_j^p(A_j, B_j, C_j, D_j) - L_k^p(A_k, B_k, C_k, D_k)] U^p(n) = 0$$

$\Leftrightarrow \forall U^p, \forall X_i(n-p), \forall X_j(n-p)$:

$$\Omega_k^p O_i^p X_i(n-p) - \Omega_k^p O_j^p X_j(n-p) + \Omega_k^p [L_i^p(A_i, B_i, C_i, D_i) - L_j^p(A_j, B_j, C_j, D_j)] = 0$$

Since Ω_k^p is defined as $\Omega_k^p O_k^p = 0$, we obtain the following result :
$Ker([O_k^p]^T) \subseteq Ker([O_i^p]^T)$ and $Ker([O_k^p]^T) \subseteq Ker([O_j^p]^T)$ and $Ker([O_k^p]^T) \subseteq Ker([L_j^p(A_j, B_j, C_j, D_j) - L_i^p(A_i, B_i, C_i, D_i)]^T)$
□

5.1.3 The mutual diagnosability property seen in the space of modes

As mentioned in Chapter 4, the space of modes offers a framework for the geometric interpretation of diagnosability, in particularly for mutual diagnosability.

Proposition 5.1 *Two modes q_i and q_j, $(i \neq j)$ are non mutually diagnosable if and only if :*
$\forall\ \zeta_{OBS}^i$ and $\forall\ \zeta_{OBS}^j$: $||R^{q_j}(\zeta_{OBS}^i)|| = ||R^{q_i}(\zeta_{OBS}^j)|| = 0$

Proof. q_i and q_j are non mutually diagnosable if and only if :
$S_{i/j} = S_{j/j} = [0, 0, ..., 0]_j^T$ and $S_{j/i} = S_{i/i} = [0, 0, ..., 0]_i^T \Leftrightarrow$
$||R_j(\zeta_{OBS}^i)|| = ||R_j(\zeta_{OBS}^j)|| = 0$ and $||R_i(\zeta_{OBS}^j)|| = ||R_i(\zeta_{OBS}^i)|| = 0 \Leftrightarrow$
$||R^{q_j}(\zeta_{OBS}^i)|| = ||R^{q_i}(\zeta_{OBS}^j)|| = 0$ □

The mutual diagnosability property can easily be interpreted geometrically in the space of modes framework, making clear what it means in terms of the regions associated to the different modes.

Theorem 5.4 *A mode q_i is mutually diagnosable from all other modes $q_j, i \neq j$, if and only if :*
$dim(\mathcal{R}_{eg}^{q_i}) = m - 1$.

Proof. ($\Rightarrow$)
$\forall q_i, \mathcal{R}_{eg}^{q_i} \subseteq \vec{q}_i^{\perp} \Rightarrow dim(\mathcal{R}_{eg}^{q_i}) \leq m-1$
In the other hand, $\forall j \neq i, q_i$ is mutually diagnosable from $q_j \Rightarrow$
$\exists \zeta_{OBS}^i$ such that $||R^{q_j}(\zeta_{OBS}^i)|| \neq 0 \Rightarrow$
$\mathcal{R}_{eg}^{q_i} \cap \{\alpha.\vec{q}_j, \alpha \in \mathbb{R}\} \neq \{\vec{0}\} \Rightarrow$
$dim(\mathcal{R}_{eg}^{q_i}) = m-1$
($\Leftarrow$)
$dim(\mathcal{R}_{eg}^{q_i}) = m-1 \Rightarrow \forall j \neq i, \exists \zeta_{OBS}^i$ such that $||R^{q_j}(\zeta_{OBS}^i)|| \neq 0$, hence q_i and q_j are mutually diagnosable. □

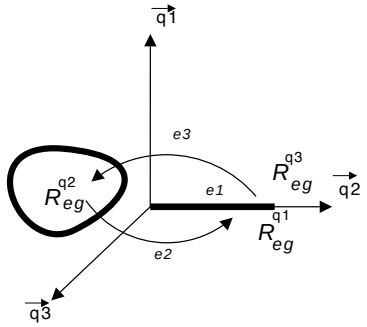

FIG. 5.2 – *Example of 2 non mutually diagnosable modes : q_1 and q_3, in the 3D-mode-space*

Proposition 5.2 *If two modes q_i and q_j are non mutually diagnosable then $dim(\mathcal{R}_{eg}^{q_i}) \leq m-2$ and $dim(\mathcal{R}_{eg}^{q_j}) \leq m-2$.*

Proof. q_i and q_j are non mutually diagnosable $\Rightarrow$
$\forall \zeta_{OBS}^i, ||R^{q_i}(\zeta_{OBS}^i)|| = ||R^{q_j}((\zeta_{OBS}^i))|| = 0 \Rightarrow$
$F(S)(n) = ||R^{q_1}||\vec{q_1} + ... + ||R^{q_i}||\vec{q_i} + ... + ||R^{q_j}||\vec{q_j} + ...||R^{q_m}||\vec{q_m} =$
$F(S)(n) = ||R^{q_1}||\vec{q_1} + ... + ||R^{q_{i-1}}||\vec{q_{i-1}} + ||R^{q_{i+1}}||\vec{q_{i+1}} + ... + ||R^{q_{j-1}}||\vec{q_{j-1}} +$
$||R^{q_{j+1}}||\vec{q_{j+1}} + ...||R^{q_m}||\vec{q_m} \Rightarrow$
$dim(\mathcal{R}_{eg}^{q_i}) \leq m-2$ and $dim(\mathcal{R}_{eg}^{q_j}) \leq m-2$ □

The Figure 5.2 shows the 3-dimension space of modes under the hypothesis that modes q_1 and q_3 are non mutually diagnosable. In this case, we notice that $\mathcal{R}_{eg}^{q_1}$ and $\mathcal{R}_{eg}^{q_3}$ are included in the vectorial subspace oriented by the vector $\vec{q}_2 : \{\alpha.\vec{q}_2, \alpha \in \mathbb{R}\}$ and $dim(\mathcal{R}_{eg}^{q_1}) = dim(\mathcal{R}_{eg}^{q_3}) = 1$.
Whereas, the mode q_2 is mutually diagnosable from all other modes, therefore $dim(\mathcal{R}_{eg}^{q_2}) = 2$.

5.2 Diagnosability of Hybrid Systems

As mentioned before, diagnosing a hybrid system consists on tracking the system mode that can be normal or faulty, by benefiting from both continuous and discrete observable behaviors. Since the mode change after the occurrence of an observable event is obviously determined, the diagnosability of hybrid systems can be formulated as follows : the hybrid system is diagnosable if and only if the occurrence of any unobservable fault event

is detected with a finite number of discrete-event and continuous observations. Since the behavior of the hybrid system is the result of underlying continuous and discrete-event behaviors, the hybrid diagnosability analysis must call upon both underlying discrete-event and continuous knowledges. To do this, the hybrid modeling framework, proposed for diagnosis in Chapter 4 is used and offers the theoretical framework for diagnosability characterization.

Properties of the hybrid language

Let us consider the hybrid language $L(S) \subseteq \Sigma^*_{hyb}$ defined in Chapter 4. As presented before, this language mixes "natural" discrete events from Σ and events issued from the abstraction of the continuous dynamics : Σ^{Sig}. Hence, some specific properties can be stated, for instance Property 5.1 illustrated in Figure 5.3.

Property 5.1 $\forall w \in L(S), w = e'.R'.w'$, *where* $e' \in \Sigma, R' \in \Sigma^{Sig}, w' \in L(S)$.

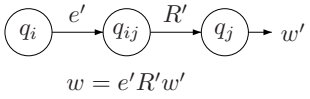

FIG. 5.3 – *Property of the hybrid language*

This property will be used later to prove the hybrid diagnosability criterion.

5.2.1 Hybrid system diagnosability definition

In this thesis, we propose a new definition of hybrid systems diagnosability based on our hybrid modeling framework and diagnosis approach.

Definition 5.7 *A fault event F is diagnosable if it can always be detected after a finite set of continuous and discrete observations i.e. after a finite sequence of observable discrete events and a finite set of continuous variable observations. The system is said to be diagnosable if and only if all the anticipated faults are diagnosable.*

This definition provides the following result in the hybrid language framework :

Proposition 5.3 *The hybrid system is diagnosable if* $\forall f$, $\exists n \in \mathbb{N}$ *such as :* $\forall s_F t \in L(S)$, *such that* s_F *ends with the occurrence of* f, *and* $t \in L(S)$ *is a continuation of* s_F *:*
$||t|| \geq n \Rightarrow (\forall w \in L(S) : P_{\Sigma_{hyb_o}}(w) = P_{\Sigma_{hyb_o}}(s_F t) \Rightarrow f \in w)$
Where $P_{\Sigma_{hyb_o}}$ *is the projection operator on the set of observable events of* Σ_{hyb} *i.e.* $\Sigma_{hyb_o} = \Sigma_o \cup \Sigma_o^{Sig}$.

5.2.2 Sufficient conditions

In this thesis, two sufficient criteria for diagnosability of hybrid systems based on the diagnosability of the underlying discrete-event and continuous systems are stated and proved.

The DES sufficient criterion

Theorem 5.5 *The hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$ is diagnosable if its underlying discrete-event system $M = (Q, \Sigma, T, q_0)$ is diagnosable.*

Proof. Given a hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$, such that the underlying discrete-event system $M = (Q, \Sigma, T, q_0)$ is diagnosable.
Given a fault $f \in \Sigma_F$ and $s_F t \in L(S)$ such that $s_F \in L(S)$ ends with the occurrence of f, and $t \in \Sigma^*_{hyb}$ is a continuation of s_F as shown in Figure 5.4.
We denote $s'_F = P_\Sigma(s_F)$ and $t' = P_\Sigma(t)$, where P_Σ is the projection on the set of discrete events Σ.
We have $s'_F \in L(M)$ ends with $f \in \Sigma_{uo} \subseteq \Sigma$, and $t' \in \Sigma^*$ is a continuation of s'_F.
Since $M = (Q, \Sigma, T, q_0)$ is diagnosable then there exists an integer n' such that : $||t'|| \geq n' \Rightarrow \forall w' \in L(M), (P_{\Sigma_o}(w') = P_{\Sigma_o}(s'_F t') \Rightarrow f \in w')$ (stated by Definition 3.2 of discrete-event system diagnosability).
We consider the integer $n = 2n' + 1$, then from Property 5.1 we have $||t|| \geq n \Rightarrow ||t'|| \geq n'$
$\forall w \in L(S)$ such that $P_{\Sigma_{hybo}}(w) = P_{\Sigma_{hybo}}(s_F t)$, we consider $w' = P_\Sigma(w)$
$P_{\Sigma_{hybo}}(w) = P_{\Sigma_{hybo}}(s_F t) \Rightarrow P_{\Sigma_o}(w') = P_{\Sigma_o}(s'_F t')$
$\Rightarrow f \in w'$ thus $f \in w$
and consequently the hybrid system S is diagnosable. □

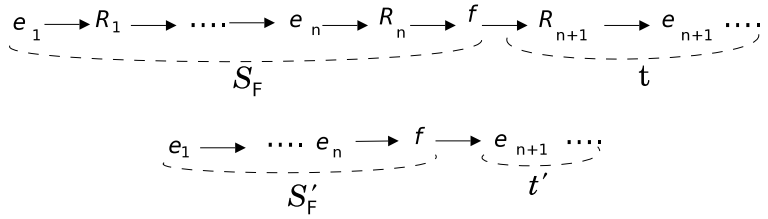

FIG. 5.4 – *The composition of a hybrid fault trajectory and its projection into the discrete-event set* Σ

The above result provides a sufficient condition for hybrid diagnosability that is solely based on the underlying discrete-event system. In practice, the underlying discrete-event system is rarely diagnosable because it does not include explicit information about the events that occur after the occurrence of a fault. The continuous knowledge is not represented and the diagnosability can only be decided on the basis of the observation of discrete control inputs and discrete sensor outputs.

The CS sufficient criterion

Theorem 5.6 *The hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$ is diagnosable if the underlying multimode system $\Xi = (\zeta, Q, C, \zeta_0)$ is diagnosable.*

Proof. Consider a hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$, such that the underlying multimode system $\Xi = (\zeta, Q, C, \zeta_0)$ is diagnosable.
Given a fault $f \in \Sigma_F$ and $s_F t \in L(S)$ such that $s_F \in L(S)$ ends with the occurrence of f as shown in Figure 5.5.

Let $q_c(q_f)$ be the mode of the system before (after) the occurrence of the fault event f.
Since the underlying multimode system is diagnosable then $\forall q_i \neq q_j, Sig(q_i) \neq Sig(q_j)$, therefore $\Sigma_{uo}^{Sig} = \emptyset$ and in addition, all the observable events $R_{o_{ij}}$ are different.
Let $t \in \Sigma^*_{hyb}$ be a continuation of s_F such that $||t|| \geq 1$.
$\forall w \in L(S)$ such that $P_{\Sigma_{hyb_o}}(w) = P_{\Sigma_{hyb_o}}(s_F t)$, Property 5.1 guarantees that $P_{\Sigma_{hyb_o}}(s_F t) = P_{\Sigma_{hyb_o}}(s_F) R_{o_{cf}} w'$ (where $w' \in \Sigma^*_{hyb_o}$).
The observation of the event $R_{o_{cf}}$ means that the system has transited from the current mode q_c to the fault mode q_f, thus $f \in w$. Hence, the hybrid system S is diagnosable. □

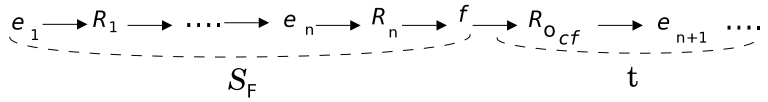

FIG. 5.5 – *Composition of a hybrid fault trajectory*

Corollary 5.1 *Two modes q_i and q_j, $i \neq j$ of the hybrid system S are diagnosable if $Sig(q_i) \neq Sig(q_j)$. If all pairs of modes (q_i, q_j), $i \neq j$ of the hybrid system are diagnosable then the hybrid system is diagnosable.*

This is again only a sufficient condition in terms of the underlying multimode system. As a matter of fact, the next section shows that continuous and discrete-event informations are required to achieve a necessary and sufficient condition.

5.2.3 The necessary and sufficient condition

As mentioned in Chapter 4, the diagnoser built from the behavior automaton (c.f. Section 4.2.6) is used, on one hand to perform on-line diagnosis, and on the other hand to check diagnosability for the hybrid system. Indeed, based on the diagnoser of the hybrid system, the diagnosability definition (Definition 5.7) is analyzed by extending discrete-event diagnosability (Theorem 3.1) to hybrid systems.
Consider $Diag(B_A(S)) = (Q_D, \Sigma_D, T_D, q_{D_0})$, the diagnoser of the hybrid system as provided in Section 4.2.7, and $\Delta_f = \{F_1, F_2, ..., F_n\}$ a set of fault labels.

Definition 5.8 *Uncertain state*
Given a diagnoser state $q_d \in Q_D$, this state is F_i-uncertain if F_i does not belong to all the labels of q_D, whereas F_i belongs to at least one label of q_D. Formally : a state $q_D \in Q_D$ is F_i-uncertain if $\exists (q, l), (q', l') \in q_D$, such that $F_i \in l$ and $F_i \notin l'$.

Definition 5.9 *Indeterminate cycle*
An F_i-indeterminate cycle in $Diag(B_A(S))$ is a cycle composed of F_i-uncertain states for which there exist two corresponding cycles in $B_A(S)$: one involves only states that carry the fault label F_i in their labels in the cycle in $Diag(B_A(S))$ and the other does not.

Proposition 5.4 *The hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$ is not F_i diagnosable if and only if the diagnoser $Diag(B_A(S)) = (Q_D, \Sigma_D, T_D, q_{D_0})$ built from the behavior automaton $B_A(S)$ contains an F_i-indeterminate cycle.*

5.2.4 Illustrative example : diagnosability analysis

Example 5.2 *Let us consider the hybrid system whose underlying discrete-event system is given in Figure 5.6. The mode signatures (that capture the continuous knowledge) are defined as follows : $Sig(N) = Sig_1$, $Sig(qF1) = Sig(qF2) = Sig_2$, $Sig(q'F1) = Sig_3$ and $Sig(q'F2) = Sig_4$. First, we focus on the diagnosability of the underlying continuous system on one hand and the discrete-event system on the other hand, which aims at checking the diagnosability of the hybrid system by means of sufficient conditions. Then, we address directly the diagnosability of the hybrid system by means of the sufficient and necessary criterion that takes into account both continuous and discrete-event knowledges.*

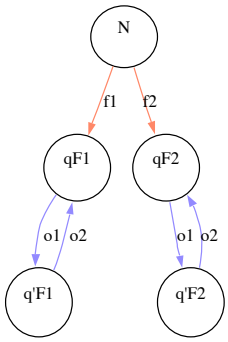

FIG. 5.6 – *Example 5.2 : the underlying discrete-event system*

Diagnosability of the underlying discrete-event system

The diagnoser of the underlying discrete-event system is given in Figure 5.7. The underlying discrete-event system is not diagnosable because of the presence of the indeterminate cycle o_1, o_2 crossing $F1$ ($F2$) uncertain states

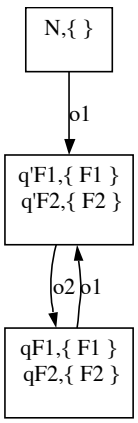

FIG. 5.7 – *Example 5.2 : the diagnoser of the underlying discrete-event system*

$\{(qF1, \{F_1\}), (qF2, \{F_2\})\}$ and $\{(q'F1, \{F_1\}), (q'F2, \{F_2\})\}$, hence, faults F_1 and F_2 are not diagnosable.

Diagnosability of the underlying continuous system

Modes $qF1$ and $qF2$ are non diagnosable because they have the same mode signature : "$Sig2$", therefore the underlying continuous system is not diagnosable w.r.t the diagnosability definition of multimode systems. Furthermore, the faults F_1 and F_2 are not diagnosable because $Sig(F_1) \cap Sig(F_2) = \{Sig_2, Sig_3\} \cap \{Sig_2, Sig_4\} = \{Sig_2\} \neq \emptyset$ (c.f. Definition 5.4). We will see later that the hybrid system may be diagnosable (for instance, faults F_1 and F_2) although the underlying continuous system is not.

Diagnosability of the hybrid system

Both discrete-event and continuous underlying systems are non diagnosable, hence the sufficient diagnosability criteria do not allow us to conclude. Hence, the necessary and sufficient criterion is required to decide about the diagnosability of the hybrid system. The behavior automaton of the hybrid system is built and provided in Figure 5.8. The diagnoser of the hybrid system is built from this beha-

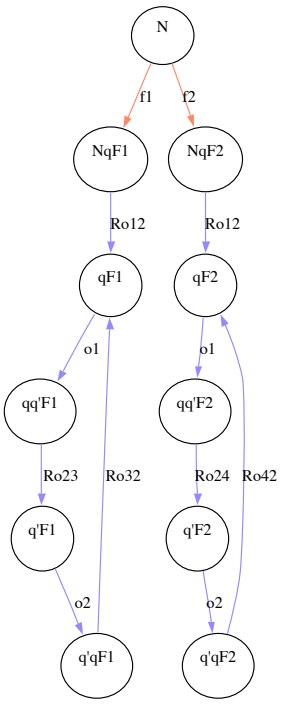

FIG. 5.8 – *Example : 5.2 : the associated behavior automaton*

vior automaton and provided in Figure 5.9. Since the hybrid diagnoser does not contain any indeterminate cycle we conclude that the hybrid system is diagnosable according to the hybrid sufficient and necessary condition (Proposition 5.4). Faults F_1 and F_2 that are non diagnosable neither in the underlying continuous system nor in the discrete-event system are diagnosable in the hybrid system.

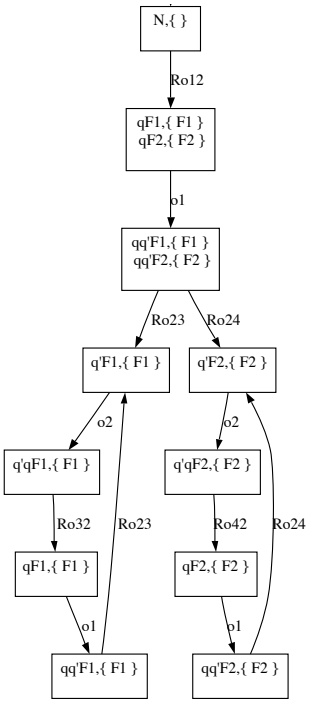

FIG. 5.9 – *Example 5.2 : the diagnoser of the hybrid system*

5.2.5 Discussion about diagnosability and mode tracking of hybrid systems

The diagnosability definition that we use is inspired from the discrete-event system definition, hence it is event-based in the sense that it is stated in terms of fault events and guarantees that a fault event is detected after a delay represented by the integer n as explained in Definition 5.3. This delay models the number of observable events (from Σ_{hyb_o}) required to detect the fault occurrence. This delay is a consequence of the disrete-event dynamics, and captures the number of mode changes needed before the fault detection. When the underlying continuous system is diagnosable, the underlying discrete-event dynamics are not required and so this delay is null. Indeed when the system modes are diagnosable, the integer n is equal to 1 for each fault f that means that all fault occurrences are detected without need of waiting for further events (the event $R_{o_{cf}}$ allows us to detect the fault as explained in proof of Theorem 5.6). Furthermore, in this case, we can precisely determine the fault mode in which the system is. However, if it is not the case, since a fault event generally corresponds to several fault modes (in which the fault is present), our event-based diagnosability formalism does not allow us to precise the fault mode. For example the diagnosability of the fault event f_1 implies that the system is in one of the fault modes associated to the fault $F1$.
As a consequence, even when the hybrid system is diagnosable, the diagnosis scheme gives only a set of *belief* modes that belong to the observable modes Q_{beh_o} (defined in Section 4.2.7). Non observable modes cannot be returned by the hybrid diagnoser module because there is not any available observable information (neither discrete-event nor continuous). Ho-

wever note that a back tracking procedure could be designed to retrieve a more precise mode history. For illustration, we propose to analyse the Example 5.3.

Example 5.3 *Consider the hybrid system whose underlying discrete-event system is provided in Figure 5.10. The mode signatures (that capture the continuous knowledge) are*

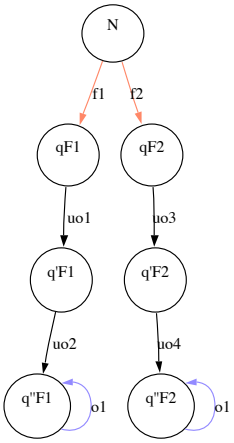

FIG. 5.10 – *Example 5.3 : the underlying discrete-event system*

defined as follows : $Sig(N) = Sig_1$, $Sig(qF1) = Sig(qF2) = Sig(q'F1) = Sig(q'F2) = Sig_2$, $Sig(q''F1) = Sig_3$ *and* $Sig(q''F2) = Sig_4$. *The behavior automaton is provided in Figure 5.11. We focus on the diagnosability of the hybrid*

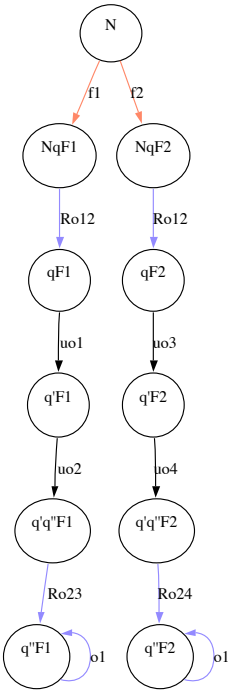

FIG. 5.11 – *Example : 5.3 : the associated behavior automaton*

system, the diagnoser of the hybrid system is so given in Figure 5.12. There are no indeterminate cycle in this diagnoser, hence the hybrid system is diagnosable. However, the diagnoser scheme presented in Chapter 4 is able to only track obser-

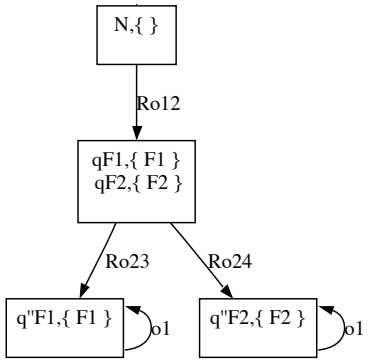

FIG. 5.12 – *Example 5.3 : the diagnoser of the hybrid system*

vable modes (Q_{beh_o}), it means modes preceded by an observable event of Σ_{hyb_o}. For illustration, let us consider the fault scenario $[f_1, uo_1, uo_2]$ that starts from the initial state "N". A comparison between Real and Estimated mode is provided in Table 5.1. We notice that faults F_1 and F_2 are diagnosable after two mode changes

Real mode	N	qF1	q'F1	q"F1
Set of estimated modes	{N}	{qF1, qF2}	{qF1, qF2}	{q"F1}

TAB. 5.1 – *Real Vs estimated mode*

$n_2 = n_3 = 2 \times 2 = 4$ (two mode changes generate 2 "natural" discrete events and 2 events from Σ^{Sig}). Modes $qF1$ and $qF2$ are observable but not diagnosable, hence, the diagnoser scheme returns the set $\{qF1, qF2\}$. However, modes $q'F1$ and $q'F2$ are non observable, hence the diagnoser scheme cannot return them. Finally, modes $q"F1$ and $q"F2$ are diagnosable and observable, hence the diagnoser scheme returns a precise diagnosis $\{q"F1\}$.

In conclusion, the main contributions provided in this chapter are the definition of diagnosability of multimode and hybrid systems, as well as the criteria for diagnosability checking. We have shown that the diagnosability of the underlying multimode or the underlying discrete-event system are only sufficient conditions, hence the sufficient and necessary criterion is provided based on the language of the hybrid system. The following chapter provides the active diagnosis scheme guided by the diagnosability properties of the system discussed in this chapter. When the system state is ambiguous, diagnosability analysis is used to decide about the control inputs able to disambiguate the ambiguous situation.

Chapter 6

Active Diagnosis Guided by Diagnosability properties

Contents

On-line diagnosis is often approached as a passive task that takes as input the available observations provided by the sensoring devices instrumenting a physical system and returns an estimation of its state, often interpreted in terms of the status of each component. However diagnosis is originally defined as a process (c.f. Hamscher *et al.* (1992)) that interlinks the determination of a belief state and the proposal of new tests that provide additional information allowing the diagnoser to refine the belief state and ultimately end with a non ambiguous state estimation. This way to go is quite common for solving post-mortem diagnosis problems, and the diagnosis is often formulated as a test sequencing problem or related in some way to testing as proposed in Struss (1994), Abramovici *et al.* (1999), Nicolaidis and Zorian (1998). The proposed tests can be achieved by :

- defining other variables to be measured
- applying other input patterns defined by specific values or signals
- driving the system to other configuration

Referring to on-line diagnosis, there are very few works mixing diagnosis and testing techniques. There are two main reasons for that : the first one is that the sensing capabilities are constrained by the available sensors, and the second is that the system's inputs are used to achieve the normal operation tasks of the system. Nevertheless, interlinking diagnosis

and test on-line, i.e. performing active diagnosis, is possible and may be necessary in some application domains, particularly those requiring autonomy. This is the case in the space domain in which the new architectures proposed for satellites are designed for giving the system self-properties. Although the constraints about on-board sensoring capabilities remain, what is different in this domain is that :

- time constraints for achieving the operating tasks are not severe and one can consider to use momentarily the inputs for diagnosis purposes.
- embedded electronic controllers acting on physical systems impose discrete switches that result in numerous configurations (or operating modes) that differ in the available measurements and may bring additional information for diagnosis.

On the other hand, active diagnosis is dictated by reliability and availability requirements. Reconfiguration actions can indeed be dangerous if the belief state is too ambiguous. The existing few works dealing with active diagnosis understood as active excitation of the system through its inputs to achieve diagnosis can be classified into two categories :

- active diagnosis for discrete-event systems : Sampath *et al.* (1998) can be mentioned as one of the only works proposing an approach for active diagnosis of discrete-event systems. A discrete-event system is modeled by a finite state machine, and active diagnosis is formulated as a supervisory control problem as in Ramadge and Wonham (1989). The novelty of the paper is to devise the controller so that specific actions that may drive the system into non diagnosable regions are forbidden. The system is hence "actively" diagnosable, allowing non ambiguous diagnosis to be performed.
- active diagnosis for continuous systems : in the field of continuous systems, Niemann (2006; 2005) are certainly the most representative works. An approach for Active Fault Diagnosis (AFD) of parametric faults is proposed for closed loop continuous systems. Auxiliary signals are introduced and a fault signature matrix in connexion with parametric faults is defined. This fault signature matrix is used for fault detection and isolation. When diagnosis based on the structure of the fault signature matrix is not possible, active diagnosis is performed thanks to the auxiliary inputs. Auxiliary input signals are designed so that the effect on system performance is minimized, but it becomes possible to detect/isolate parametric faults in the system.

Conditional planning for active diagnosis

This thesis addresses active diagnosis problem of hybrid systems. The passive diagnosis approach presented in Chapter 4 does not take into account the diagnosability properties of the system. Therefore, the returned diagnosis may be ambiguous in the sense that the mode tracking process returns more than one possible current mode, jeopardizing in this way, the reconfiguration process. Therefore, in the autonomy context, the system must be able to autonomously leave such an ambiguous situation by actively diagnosing the system mode. Active diagnosis is introduced as the solution of this problem (c.f. Bayoudh *et al.* (2008c)). Indeed, starting

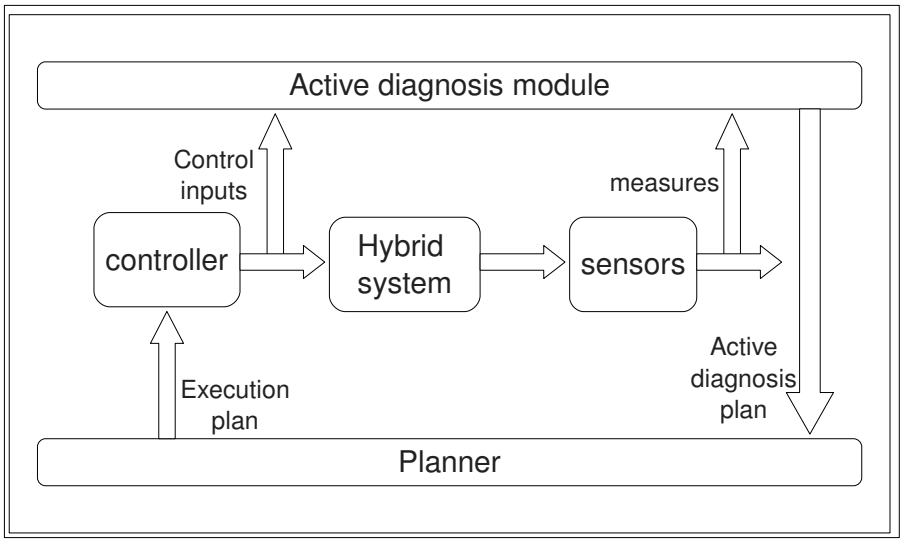

FIG. 6.1 – *The active diagnosis scheme for hybrid systems*

from an ambiguous situation, the active diagnosis process consists of performing additional control inputs to change the system configuration in order to exhibit further diagnosis information. Therefore, active diagnosis must be seen as an integrated control and diagnosis problem. The active diagnosis objectives must be consistent with the normal operation control objectives. The system's operation is required, in the worst case, to remain safe and, in the best case, to preserve normal performances when specific control inputs are applied to drive the system into state space regions that exhibit the appropriate symptoms. Hence, active diagnosis actions must be optimized and those that can be dangerous for the system must be forbidden.

This chapter provides a scheme to achieve active diagnosis in our hybrid system framework. Starting with an ambiguous belief state[1] , our method calls for diagnosability analysis results to determine a new system configuration in which fault candidates can be discriminated. The control inputs to be applied to the system to drive it into this configuration are determined, paying attention to avoid states that could be dangerous for the system. In our modeling framework, the system behavior after a fault occurrence is modeled by an anticipated fault mode with associated continuous dynamics. Possible control actions after the occurrence of a fault are modeled by transitions outgoing the corresponding faulty mode. These control actions are the key of active diagnosis. Seeing that mode transitions can be resulting from discrete-event or continuous dynamics evolution, both continuous and discrete control actions can be used, in an interlinked way, to perform active diagnosis. These interlinked actions ultimately act by putting the system in a goal configuration, i.e. a goal behavioral mode, by driving it through a selected sequence of intermediary behavioral modes. In contrast to Sampath *et al.* (1998) that forbids non diagnosable regions with appropriate control actions, our approach is based on driving the system towards diagnosable regions. These regions

[1] An ambiguous belief state (or situation) of the system corresponds to an uncertain state of the diagnoser of the hybrid system.

correspond to non uncertain states of the hybrid system diagnoser and represent the *target states* of the active diagnosis problem.
This chapter starts by defining the active diagnosis problem of hybrid systems, the new concepts of controllable and induced controllable events are introduced. Based on these concepts, a new finite state machine called *the active diagnoser* is defined (c.f. Bayoudh *et al.* (2008c; 2009)) in order to on-line perform active diagnosis.

6.1 Defining the active diagnosis problem for hybrid systems

Let us assume that a hybrid system is continuously monitored and that its state is tracked following the passive diagnosis approach proposed in Chapter 4. Assume that the current belief state returned by the diagnoser is faulty and uncertain, i.e. several faults are candidate. This is the starting point of an active diagnosis session. The active diagnosis problem is formulated as a conditional planning problem. From an uncertain state of the diagnoser, the plan defines how to find a controllable paths leading to a certain state. The search of active diagnosis actions is guided by the observable response of the system on active control inputs.
What is important to notice is that even when the conditions for non diagnosability as stated by Proposition 5.4 hold, there may be a way to enforce a sequence of transitions to drive the system towards a non certain state of the diagnoser. Indeed, an indeterminate cycle of the diagnoser only indicates that the system may get stuck in the cycle. Therefore, we distinguish two situations for which an active diagnosis session is triggered :

- the uncertain state belongs to an indeterminate cycle, in this case the system is non diagnosable w.r.t this state and the active diagnosis aims at cutting the indeterminate cycle and bringing the diagnoser in a certain state.
- the uncertain state does not belong to an indeterminate cycle, in this case the system is diagnosable w.r.t this state, however, the active diagnosis aims at energizing the diagnoser to leave this state (the system does not wait for observations, the controller sets them off).

6.2 Introducing the notion of controllable and induced controllable paths

Active diagnosis is closely linked to the property of controllability of the system. Indeed, the active diagnosis consists on determining paths from the starting uncertain state of the active diagnoser to target states in which the diagnosis is precise (or more precise). Consequently the dynamics of the system along these paths must be controllable to allow the system to be driven to the target states. Hence, concepts of *controllable events* and *induced controllable events* are introduced.
Let us consider the hybrid language $L(S) \subseteq \Sigma_{hyb}^*$ and let us call $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_{hyb}$ the set of controllable events[2].

Definition 6.1 *Controllable event*
Controllable events fall in one of the categories below :

- *discrete control inputs, $c \in \Sigma_c$ (for example, the software commands sent by embedded calculators).*
- *events, $\sigma_u \in \Sigma_c$, corresponding to spontaneous mode changes when the continuous dynamics model of the source mode is controllable in the sense of Terrell (1999). This means that there always exists a continuous control law u that leads to the occurrence of such event.*

[2]Controllable actions are assumed to be observable.

The set of possible transitions outgoing fault modes represents all the control actions that can be done to perform active diagnosis. The set of these *allowed* control actions is different for the different fault modes, and is a mean to account for safety constraints.

Definition 6.2 *Induced controllable event*
Events whose occurrence always follow the occurrence of a controllable event are called induced controllable events and form the set $\Sigma_{hyb_{ic}} \subseteq \Sigma_{hyb}$.

Induced controllable events model the response of the hybrid system after a control action either a discrete input event or a continuous input signal. Induced controllable events fall in one of the categories below :

- $\Sigma_{ic}^{Sig} \subseteq \Sigma^{Sig}$: the set of induced controllable events that manifest the reaction of the continuous dynamics. Let $R_{ij} \in \Sigma^{Sig}$ denote a discrete event associated to a mode signature change. R_{ij} is an induced controllable event denoted R_{ij}^{ic} if the mode change is controlled by a controllable event.
- $\Sigma_{ic} \subseteq \Sigma$: the set of induced controllable events that manifest the reaction of discrete dynamics. $\sigma \in \Sigma_{ic}$ is an induced discrete event denoted σ_{ic} if its occurrence is always a consequence of a given controllable event.

The set of induced controllable events of the hybrid system is given as $\Sigma_{hyb_{ic}} = \Sigma_{ic} \cup \Sigma_{ic}^{Sig}$. Controllable events are those that provide means to act on the system. Induced controllable events are those that manifest the reaction of the system and allow us to discriminate ambiguous situations.

Definition 6.3 *Controllable path*
Consider the hybrid system behavior automaton and its associated hybrid language $L(S) \subseteq \Sigma_{hyb}^{*}$[3]. *A controllable path s is a string of controllable and induced controllable events,* $s \in (\Sigma_c \cup \Sigma_{hyb_{ic}})^*$. *Formally, a controllable path is* $s = \alpha_1\beta_1, ..., \alpha_k\beta_k$, *with* $\alpha_i \in 2^{\Sigma_c}$ *and* $\beta_i \in \Sigma_{hyb_{ic}}$, $i = 1..k$, $k \in \mathbb{N}^*$.

A controllable path in the behavior automaton corresponds to a controllable observable path in the corresponding diagnoser.

6.3 Towards an active diagnoser

The idea proposed in this thesis is to use the diagnoser to guide the search for the sequence of actions that will disambiguate an ambiguous belief state in the diagnoser. However, in order to suit active diagnosis purposes, the diagnoser must be modified into an *Active Diagnoser* that involves only controllable paths. Classically, the control actions that appear in the diagnoser are supposed to be observed but not necessarily applied. In particular, a control event associated to a transition outgoing an uncertain state of the diagnoser is only observed in *at least* one of the underlying faulty modes of the system. In our case, we want to actively apply the control event, which means that it must be applicable in *all* the faulty modes included in the concerned diagnoser state, otherwise it means that the control

[3] As defined in Section 4.2.6 of Chapter 4.

is forbidden as it may be dangerous in some underlying modes. The diagnoser is hence modified accordingly.
Given an uncertain state of the diagnoser, outgoing transitions associated with controllable events are removed if there is no corresponding transition outgoing from *all* the corresponding modes of the behavior automaton.
In our modeling, all enabled control inputs in faulty modes are represented in the mode automaton and can be used to perform active diagnosis. Control inputs that do not appear in the mode automaton must be forbidden and can be dangerous for the system (this is achieved by Equation 6.1).
Let us consider the example shown in Figure 6.2 (left). When the system is in the faulty mode $qF2$, the set of enabled control inputs is $\{c1, c2\}$. However, in the faulty mode $qF1$, the set of enabled control inputs is reduced to $\{c1\}$. It means that the control input $c2$ is forbidden in mode $qF1$. Now,

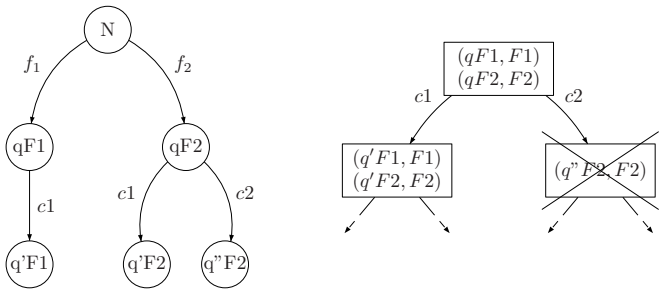

FIG. 6.2 – *Enabled actions for active diagnosis*

let us consider the associated diagnoser shown in Figure 6.2 (right). Transitions outgoing the uncertain diagnoser state $\{(qF1, \{F1\}), (qF2, \{F2\})\}$ are labeled by control inputs $c1$ and $c2$. However, by considering the mode automaton, only $c1$ can be enabled. Hence, $c2$ must be removed.
Formally the construction of the active diagnoser is achieved from $Diag(B_A(S))$ (c.f. Chapter 4, Section 4.2.7) by defining a new partial transition function $T_D^{act} \subseteq Q_D \times \Sigma_{hyb_o} \rightarrow Q_D$ as follows, where two cases are distinguished :

- $\sigma \in \Sigma_c$:

$$T_D^{act}(q_D, \sigma) = \bigcup_{(q,l)\in q_D} \{(T_{beh}(q, \sigma), l)\} \text{ if } (\forall (q, l) \in q_D, T_{beh}(q, \sigma) \neq \emptyset),$$
$$\text{otherwise } T_D^{act}(q_D, \sigma) = \emptyset \quad (6.1)$$

- $\sigma \in \Sigma_{hyb_o} \setminus \Sigma_c$:

$$T_D^{act}(q_D, \sigma) = \bigcup_{\substack{(q,l)\in q_D \\ s\in L_\sigma(S,q)\cap\Sigma^*_{hyb_{ic}}}} \{(T_{beh}(q, s), LP(q, l, s))\} \quad (6.2)$$

6.4 Conditional planning for determining an active diagnosis plan

As mentioned before, active diagnosis consists on exciting the hybrid system to exhibit additional observations. Given an uncertain state of the

active diagnoser, the active diagnosis problem is how to find controllable paths leading to certain states. In the uncertain state, the active diagnosis is performed by triggering a sequence of consecutive controllable events, observing the system reaction, and deciding about the next sequence. The choice of the consecutive controllable event sequence depends on the last observed induced controllable event. This problem is formulated as a conditional planning in a full observable environment problem (c.f. Bertoli *et al.* (2001), Jimenez and Torras (2000), Russel and Norvig (2003)). The active diagnoser is seen as an *AND-OR* graph. The "OR" nodes (squares) correspond to the selection of a possible sequence of consecutive controllable events, the "AND" nodes (circles) correspond to the resulting induced events as shown in Figure 6.3. A modified *MINIMAX* algorithm (c.f. Algorithm 1) is applied to resolve the conditional planning problem Bayoudh and Travé-Massuyès (2009). The algorithm is performed from an uncertain state and searches all controllable paths leading to certain states. The active diagnosis session can be started only from an uncertain state that belongs to the active diagnoser ($q_D \in T_D^{act}(Q_D, \Sigma_{hyb_o})$).

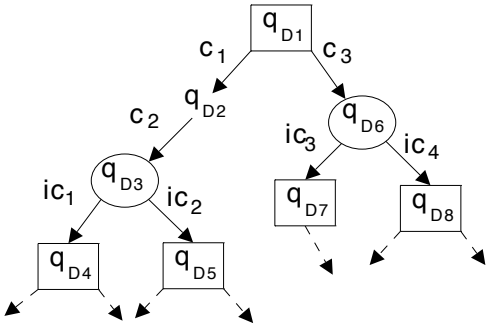

FIG. 6.3 – *The active diagnosis seen as a planning problem*

6.4.1 Conditional planning algorithm

The mapping between the active diagnoser and the *AND-OR* graph is described as follows :

- the state nodes (OR nodes) S_k of the graph correspond to the states q_{D_k} of the active diagnoser (represented by squares in Figure 6.3) in which sequences of controllable events are started (for example, states q_{D_1}, q_{D_4}, q_{D_5}, q_{D_7} and q_{D_8} of the active diagnoser shown in Figure 6.3).
- actions $a_i \in 2^{\Sigma_c}$ are the sequences of consecutive controllable events starting in state nodes (for example, in Figure 6.3, $a_1 = [c_1, c_2]$ and $a_2 = [c_3]$).
- the observation nodes O_j (AND nodes) of the graph (represented by circles in Figure 6.3) correspond to the state q_{D_j} of the active diagnoser in which the outgoing transitions are labeled with induced controllable events (for example, states q_{D_3} and q_{D_6} of the active diagnoser shown in Figure 6.3). An observation o_k outgoing an observation node O_j corresponds to an induced controllable event $\sigma_{ic_k} \in \Sigma_{hyb_{ic}}$ and leads to a next state node (for example $o_1 = ic_1$

and $o_2 = ic_2$ that are associated to the observation node $O_1 = q_{D_3}$ as well as $o_3 = ic_3$ and $o_4 = ic_4$ that are associated to the observation node $O_2 = q_{D_6}$, in the active diagnoser shown in Figure 6.3). We link in a pair (S_k, o_k) the state node S_k with the observation o_k that corresponds to the induced controllable discrete event leading this state (for example $(S_1, o_1) = (q_{D_1}, \emptyset)$, $(S_2, o_2) = (q_{D_4}, ic_1)$, $(S_3, o_3) = (q_{D_5}, ic_2)$, $(S_4, o_4) = (q_{D_7}, ic_3)$ and $(S_5, o_5) = (q_{D_8}, ic_4)$ in the active diagnoser shown in Figure 6.3[4]).

- target states of the graph correspond to certain states of the active diagnoser. Let us assume that the active diagnosis session is started in a diagnoser state uncertain with respect to every fault F_{i_j} in a set $\mathcal{F} = \{F_{i_1}, F_{i_2}, ..., F_{i_n}\}$, i.e. F_{i_1}-uncertain, F_{i_2}-uncertain, ..., and F_{i_n}-uncertain state. Then the set of target states is composed by F_{i_1}, F_{i_2}, ..., and F_{i_n} certain states (i.e. $\mathcal{F}$-certain states) and denoted $\Delta_{certain}$. This set can be relaxed to a set of $2^{\mathcal{F}}$ certain states when the active diagnosis is not expected to achieve single fault diagnosis refinement.
- the initial state of the graph is a state node S_1 that corresponds to an uncertain state of the active diagnoser in which the active diagnosis session is started.

Notice that in the active diagnoser shown in Figure 6.3 the state q_{D_2} is neither an AND node, nor an OR node because it is preceded and followed by a controllable event.

We define the SUCCESSORS function that for each pair (S_k, o_k) of node state and linked observation, associates an action a outgoing S_k and a set of corresponding successor node states (and their associated observations) : $\bigcup_k \{(S_{k'}, o_{k'})\}$.

A modified *MINIMAX* algorithm is proposed for the AND-OR graph exploration (Algorithm 1). For conditional planning the minimax algorithm is modified as follows. First MAX and MIN nodes become OR and AND nodes. The plan needs to take some action at every state it reaches, but must account for every observation after an action is taken (c.f. Russel and Norvig (2003)). Second, the algorithm needs to return a conditional plan rather than just a single action. At an OR node, the plan is just the action selected, followed by whatever comes next. At an AND node, the plan is a nested series of if-then-else steps specifying subplans for each possible outcome, the tests in these steps being the associated state observations. More details are provided in Russel and Norvig (2003).

The algorithm is a recursive depth-first algorithm, an important point is that it deals with cycles, which often arise in non diagnosable system diagnosers. Indeed, when the current state is identical to a state on the path from the root, then it returns failure. This does not mean that there is no solution from the current state, but simply means that if there is one, it must be reachable from the earlier instance of the current state, so the new instance can be discarded. With this check, we ensure that the algorithm always terminates (the state space that is a part of the active diagnoser is finite) (c.f. Russel and Norvig (2003)).

[4]Notice that the observation associated to the starting state of the active diagnosis session is the empty element.

Algorithm 1 : AND-OR search algorithm for active diagnosis

```
FUNCTION AND-OR-GRAPH-SEARCH ( ) ;
returns a conditional plan, or failure ;
begin
    OR-SEARCH((S_0, ∅), [ ] )
end
FUNCTION OR-SEARCH ((S,o), path) ;
returns a conditional plan, or failure ;
begin
    if S = certain-state then
        return the-empty-plan
    if S ∈ path then
        return failure
    foreach a, sate-observation-set ∈ SUCESSORS((S, o)) do
        plan ⟵ AND-SEARCH(sate-observation-set, [S | path]
    if plan ≠ failure then
        return [a | plan]
    return failure
end
FUNCTION AND-SEARCH (state-observation-set, path) ;
returns a conditional plan, or failure ;
begin
    foreach (S_i, o_i) ∈ state-observation-set do
        plan_i ⟵ OR-SEARCH((S_i, o_i), path)
    if plan_i = failure then
        return failure
    return  [if o_1 then
                plan_1 ;
                else if o_2 then
                    plan_2 ;
                    else if ... then
                        ... ;
                        else if o_{n-1} then
                            plan_{n-1} ;
                            else
                                plan_n]
end
```

6.4.2 Discussion

Algorithm 1 explores the AND-OR graph corresponding to the active diagnoser and returns all controllable paths leading to certain states. Each path is a conditional plan for the active diagnosis. A plan can be then executed by the controller. Two types of plans can be distinguished :

Definition 6.4 *Guaranteed plan*
A conditional plan is said to be guaranteed *if it guarantees to reach a certain state of the active diagnoser from the starting uncertain state.*

Indeed, a guaranteed plan anticipates all the possible resulting induced controllable events following an action included in the plan. In the opposite, the plan is not guaranteed if it contains at least one action for which at least one possible resulting induced controllable event is not anticipated by the plan. When we execute a guaranteed plan, we have the guaranty that the system will reach a target state (a certain state) because all possible resulting situations after an action are taken into account.
In the contrary, when we execute an plan that is not guaranteed, the reachability of a certain state is not guaranteed. After an action, if an induced controllable event that is not anticipated occurs, the plan fails. When there is no guaranteed plan, the system must be able to choose the best plan among the non guaranteed plans. Costs as well as probabilities can be associated to the control actions in order to help with the decision. In this case, the AND-OR graph exploration could be achieved by AO^* type algorithms based on heuristic search.

6.4.3 Diagnosability and active diagnosis

This section addresses the link between active diagnosis and diagnosability.

Definition 6.5 *Active diagnosability*
The hybrid system is actively diagnosable if for any uncertain state of the diagnoser a guaranteed *plan exists in the active diagnoser which starts from the uncertain state and leads to a certain state.*

Definition 6.5 ensures that the system controller is able to bring the diagnoser out of any uncertain state. This definition is different from the diagnosability definition (c.f. Definition 5.7), in the sense that the definition of active diagnosability takes into account not only the observation system, but also the system controller properties.
A relaxed definition of active diagnosability called *non-guaranteed active diagnosability* is now proposed :

Definition 6.6 *Non-guaranteed active diagnosability*
The hybrid system is actively diagnosable if for any uncertain state of the diagnoser there exists a plan (guaranteed or non guaranteed) in the active diagnoser, which starts from this uncertain state and leads to a certain state.

In the following, we extend the definition of I-diagnosability from Sampath *et al.* (1995) recalled by Definition 3.5 to hybrid systems to establish a link between I-diagnosability and active diagnosability of hybrid systems.

First we define $\Sigma_I \subseteq \Sigma_o$ the set of indicator events. Then, for each fault event $f \in \Sigma_F$ we associate a set of indicator events $I_f(f) \subseteq \Sigma_I$ defined by the indicator map $I_f : \Sigma_F \rightarrow 2^{\Sigma_I}$.

Definition 6.7 *I-diagnosability of hybrid systems*
The hybrid system is I-diagnosable w.r.t I_f if $\forall f \in \Sigma_F$, $\exists n \in \mathbb{N}$ such as : $\forall s_F t \in L(S)$, such that s_F ends with the occurrence of f, and $t \in L(S)$ is a continuation of s_F such that $I_f(f) \subseteq P_{\Sigma_I}(t)$:
$||t|| \geq n \Rightarrow (\forall\, s \in L(S) : P_{\Sigma_{hyb_o}}(s) = P_{\Sigma_{hyb_o}}(s_F t) \Rightarrow f$ occurs in s),
where $P_{\Sigma_{hyb_o}}$ is the projection operator on the set of observable events and P_{Σ_I} is the projection operator on the set of indicator events.

Sampath *et al.* (1995) proves that a fault f is non I-diagnosable if and only if there exists an indeterminate cycle following the occurrence of the associated indicator events included in $I_f(f)$.
Let us analyze how this can be interpreted in the active diagnosis context. The a set of indicator events $I_f(f)$ associated to each fault f is now assumed to be composed by controllable events belonging to any active diagnosis plan (guaranteed or not) starting with an F-uncertain state and ending with an F-certain state. Link between actively diagnosability and I-diagnosability with respect to active diagnoser can is established.

Proposition 6.1 *If the hybrid system is actively diagnosable(in a guaranteed or not guaranteed way) then it is I-diagnosable with respect to I_f (the reciprocal is not true [5]).*

Proof. If the hybrid system is actively-diagnosable, then the set of active diagnosis plans returned by Algorithm 1 started from a any *F*-uncertain state of the diagnoser is not empty.
Let us take $I_f(f)$ as the set of controllable events belonging to any active diagnosis plan (guaranteed or not). The system is obviously I-diagnosable w.r.t the map indicator I_f. □

6.5 Illustrative example

Let us consider a hybrid system consisting of three tanks of water, $T1$, $T2$ and $T3$. Valves $V1$ and $V2$ allow the flow to transfer between tanks. Valves

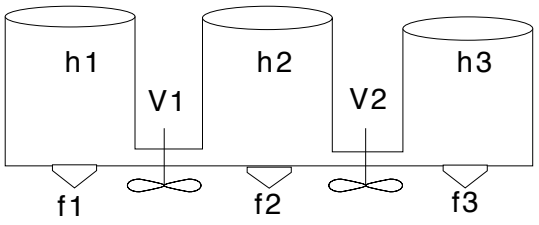

Fig. 6.4 – *The three-tanks system*

are controlled by discrete control inputs $open_{V_1}$, $open_{V_2}$, $close_{V_1}$ and $close_{V_2}$.

[5]The reciprocal is not true because I-diagnosability does not account for the controllability of paths outgoing the concerned uncertain state, which is present in the active diagnosis plans.

The system is equipped with three level sensors that measure the level of water in each tank. Hence, water levels h_1, h_2 and h_3 are observable.
The discrete behavior of the system between nominal modes (no fault) is described in Figure 6.5.
Every nominal mode models a configuration of the system as shown in

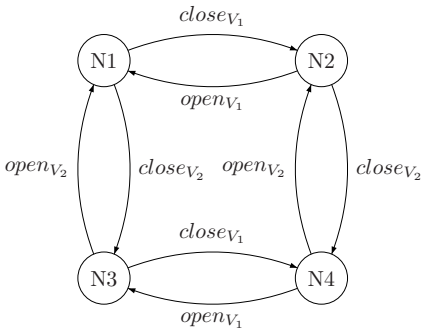

FIG. 6.5 – *The mode automaton of the nominal behavior of the three tanks system*

Table 6.1.
Fault events f_1, f_2 and f_3 model leaks that may occur in tanks $T1$, $T2$

Nominal mode	$N1$	$N2$	$N3$	$N4$
Valve V1	*opened*	*closed*	*closed*	*opened*
Valve V2	*opened*	*opened*	*closed*	*closed*

TAB. 6.1 – *The system configuration in nominal modes*

and $T3$, respectively. A fault event $f_j, 1 \leq j \leq 3$ may occur in any nominal mode $N1, N2, N3$ and $N4$ and leads to anticipated fault mode $1Fj$, $2Fj$, $3Fj$ and $4Fj$, respectively. This is shown in Figure 6.6.
The observable continuous behavior in every mode (nominal or faulty) is

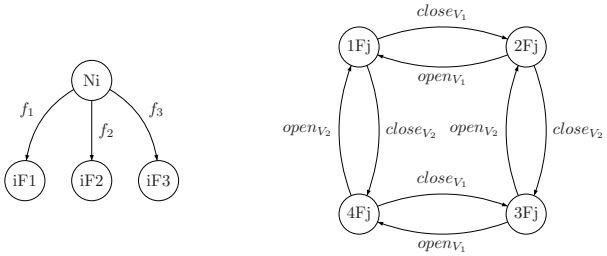

FIG. 6.6 – *The anticipated fault modes of the three-tanks system*

described by constraints linking observable variables given in Table 6.2.
Boolean consistency indicators (residuals) $r_i, i = 1..6$, are associated to every constraint and allow one to check the consistency between observations and system model (see Table 6.3).
For sake of clarity, shared constraints are considered only once in the mode signatures of the system. Given $[r_1, r_2, r_3, r_4, r_5, r_6]$ the vector of all system residuals, the mode signature is computed on-line by evaluating this vector using system observations. The mode theoretical signatures of the system are given in Table 6.4.

$N1, N2, N3, N4$	$\frac{dh_1}{dt} = 0, \frac{dh_2}{dt} = 0, \frac{dh_3}{dt} = 0$
$1F1, 1F2, 1F3$	$\frac{dh_1}{dt} < 0, \frac{dh_2}{dt} < 0, \frac{dh_3}{dt} < 0$
$2F1, 3F1$	$\frac{dh_1}{dt} < 0, \frac{dh_2}{dt} = 0, \frac{dh_3}{dt} = 0$
$2F2, 2F3$	$\frac{dh_1}{dt} = 0, \frac{dh_2}{dt} < 0, \frac{dh_3}{dt} < 0$
$3F2$	$\frac{dh_1}{dt} = 0, \frac{dh_2}{dt} < 0, \frac{dh_3}{dt} = 0$
$3F3, 4F3$	$\frac{dh_1}{dt} = 0, \frac{dh_2}{dt} = 0, \frac{dh_3}{dt} < 0$
$4F1, 4F2$	$\frac{dh_1}{dt} < 0, \frac{dh_2}{dt} < 0, \frac{dh_3}{dt} = 0$

Tab. 6.2 – *Set of continuous constraints in each operating mode*

$\frac{dh_1}{dt} = 0 \Leftrightarrow r_1 = 0$	$\frac{dh_3}{dt} = 0 \Leftrightarrow r_3 = 0$	$\frac{dh_2}{dt} < 0 \Leftrightarrow r_5 = 0$
$\frac{dh_2}{dt} = 0 \Leftrightarrow r_2 = 0$	$\frac{dh_1}{dt} < 0 \Leftrightarrow r_4 = 0$	$\frac{dh_3}{dt} < 0 \Leftrightarrow r_6 = 0$

Tab. 6.3 – *The consistency indicators*

Let's consider the case when any of the fault events $f1$, $f2$ or $f3$ occur in

$Sig(N1) = Sig(N2) = Sig(N3) = Sig(N4) = \begin{pmatrix}0\\0\\0\\1\\1\\1\end{pmatrix}$
$Sig(2F1) = Sig(3F1) = \begin{pmatrix}1\\0\\0\\0\\1\\1\end{pmatrix}$, $Sig(2F2) = Sig(2F3) = \begin{pmatrix}0\\1\\1\\1\\0\\0\end{pmatrix}$
$Sig(1F1) = Sig(1F2) = Sig(1F3) = \begin{pmatrix}1\\1\\1\\0\\0\\0\end{pmatrix}$, $Sig(3F2) = \begin{pmatrix}0\\1\\0\\1\\0\\1\end{pmatrix}$
$Sig(3F3) = Sig(4F3) = \begin{pmatrix}0\\0\\1\\1\\1\\0\end{pmatrix}$, $Sig(4F1) = Sig(4F2) = \begin{pmatrix}1\\1\\0\\0\\0\\1\end{pmatrix}$

Tab. 6.4 – *Mode Signatures of the three-tanks system*

the nominal mode $N1$. The corresponding partial behavior automaton is shown in Figure 6.7.

Events R^{ic}_{o1}, R'^{ic}_{o1}, R^{ic}_{o2}, R'^{ic}_{o2}, R^{ic}_{o3}, R'^{ic}_{o3}, R^{ic}_{o4} and R'^{ic}_{o4} (c.f. Table 6.5) correspond to the observable switches of the mode signature that follow the control inputs. They belong to the set of induced controllable events $\Sigma_{hyb_{ic}}$. R_{of} corresponds to the observable mode signature switch after the occurrence of any of the fault events $f1$, $f2$ or $f3$. As previously mentioned, non observable signature switches (Σ^{Sig}_{uo}) are not considered.

The diagnoser of the three-tanks system is computed from the behavior automaton. Let us focus on the part of the active diagnoser shown in Figure 6.8. The occurrence of the fault event $f1$, $f2$ or $f3$ is detected by the observation of the observable event R_{of}. The presence of the indeterminate cycle $[\{(\{2F2, \{F2\}), (2F3, \{F3\})\}, \{(\{21F2, \{F2\}), (21F3, \{F3\})\}, \{(\{1F2, \{F2\}), (1F3, \{F3\})\}, \{(\{12F2, \{F2\}), (12F3, \{F3\})\}]$ (cycle defined by the red transitions in Figure 6.8) proves (c.f. Proposition 5.4 of Chapter 5) that the language of the hybrid system is not diagnosable.

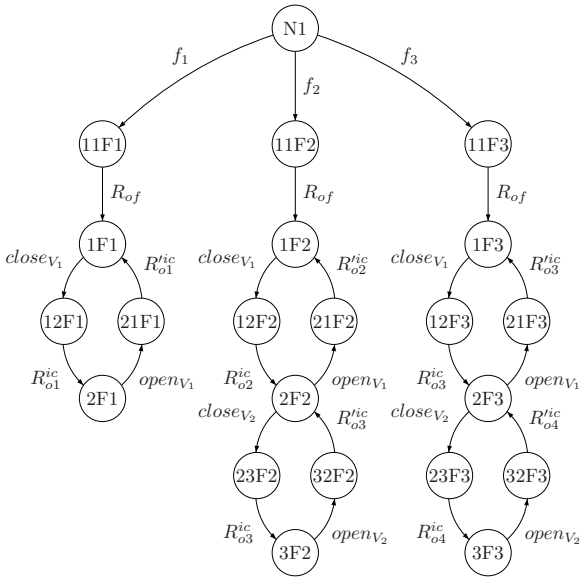

FIG. 6.7 – *Part of the behavior automaton of the three-tanks system*

The non diagnosability of the system language is due to the non diagnosability of faults $f2$ and $f3$ pointed out by the indeterminate cycle shown in Figure 6.8. However, we show that performing active diagnosis allows us to diagnose the system with certainty.
The active diagnosis consists in searching a conditional plan that permits to leave the starting uncertain state of the diagnoser and reach a certain state. These uncertain states may be crossed by indetermi-

Source mode	Destination mode	Associated event
$N1$	$\{1F1, 1F2, 1F3\}$	R_{of}
$1F1$	$2F1$	R^{ic}_{o1}
$\{1F2, 1F3\}$	$\{2F2, 2F3\}$	R^{ic}_{o2}
$2F2$	$3F2$	R^{ic}_{o3}
$2F3$	$3F3$	R^{ic}_{o4}
$2F1$	$1F1$	R^{ric}_{o1}
$\{2F2, 2F3\}$	$\{1F2, 1F3\}$	R^{ric}_{o2}
$3F2$	$2F2$	R^{ric}_{o3}
$3F3$	$2F3$	R^{ric}_{o4}

TAB. 6.5 – *Observable events associated to mode signature changes*

nate cycles (example : $\{(\{2F2, \{F2\}), (2F3, \{F3\})\}$) or not (example : $\{(1F1, \{F1\}), (1F2, \{F2\}), (1F3, \{F3\})\}$).
The active diagnoser of the three-tanks system is seen as a *AND-OR* graph as explained in Section 6.4.1. From the uncertain state of the diagnoser the active diagnosis plan is given by the *MINIMAX* algorithm. An active diagnosis plan defines a set of controllable paths from the uncertain state of the diagnoser to certain states.
Given the system diagnoser, the occurrence of any fault event $f1, f2$ or $f3$ is detected by the observable events R_{of} and puts the diagnoser in the

uncertain state $\{(1F1,\{F1\}),(1F2,\{F2\}),(1F3,\{F3\})\}$. From this uncertain state the active diagnosis plan is : $[close_{V_1},\ if\ R^{ic}_{o2}\ close_{V_2}\ Else\ [\ ]\ ]$.
Consequently, to perform active diagnosis, the controller sends the discrete-control-input $close_{V_1}$, if the resulting observed induced controllable event is R^{ic}_{o2} (i.e. the diagnoser state is $\{(2F2,\{F2\}),(2F3,\{F3\})\}$) then it sends the control input $close_{V_2}$ to discriminate between $F2$ and $F3$, else, the resulting observed induced controllable event is R^{ic}_{o1} (i.e. the diagnoser has reached the certain state $\{(2F1,\{F1\})\}$ and the controller does not send any more discrete control input.
Let us notice that the plan $[close_{V_1},\ if\ R^{ic}_{o2}\ close_{V_2}\ Else\ [\ ]\ ]$ is a guaranteed plan, because after the action $close_{V_1}$ ($close_{V_2}$), the possible resulting induced controllable events R^{ic}_{o1} and R^{ic}_{o2} (R^{ic}_{o3} and R^{ic}_{o4}) are anticipated by the plan.

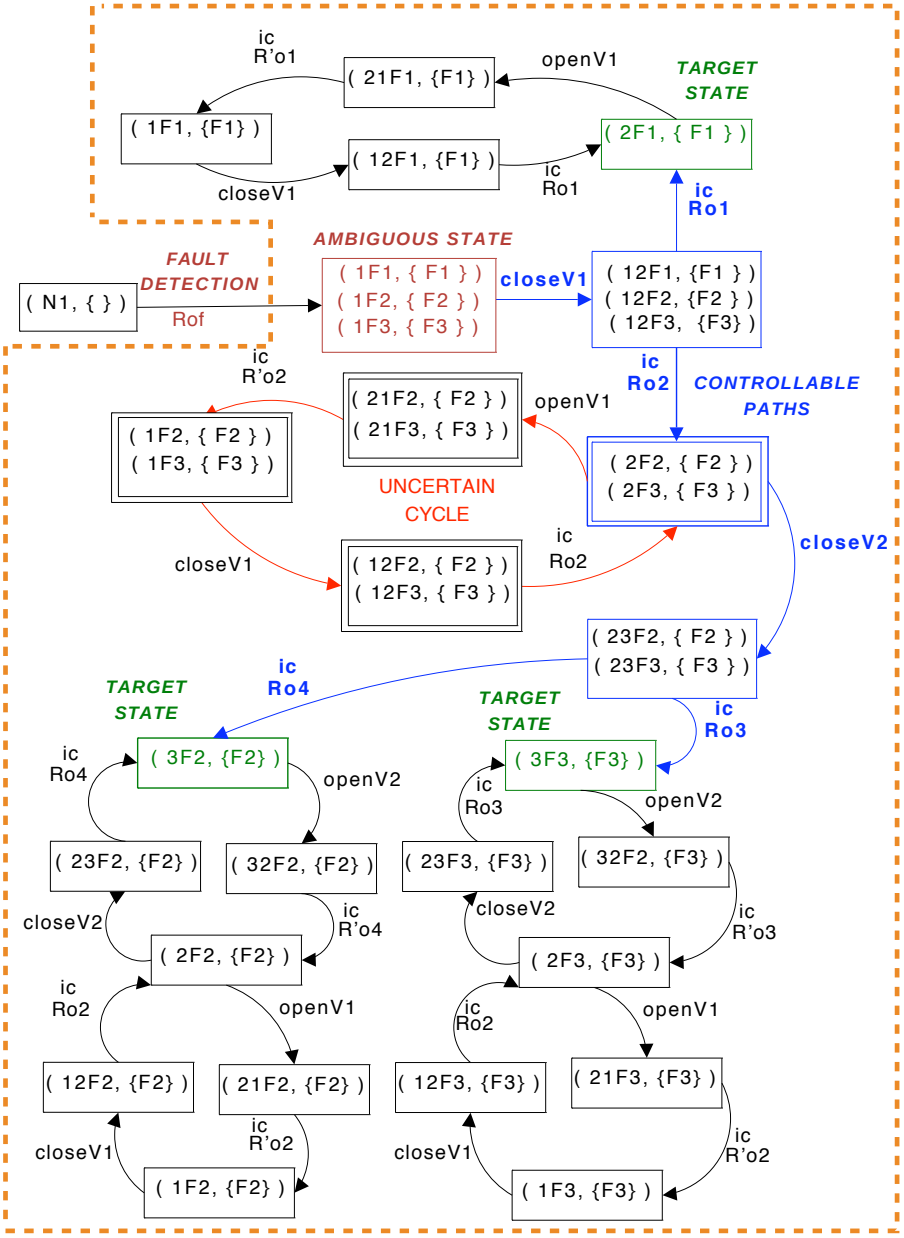

FIG. 6.8 – *The active diagnoser of the three-tanks system*

Chapter 7
The Attitude Control System (ACS) with Reaction Wheels

Contents

The spacecraft *attitude* describes its orientation compared to external reference frames and is influenced by external disturbances (torques exerted on the spacecraft). Therefore a control system is necessary for sys-

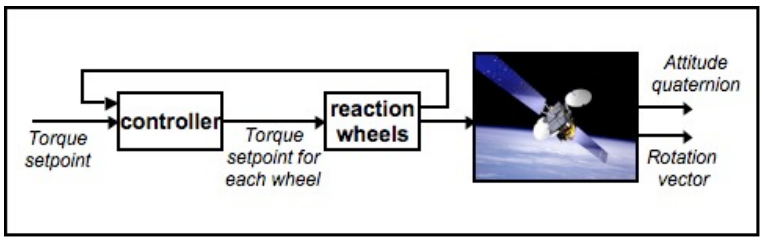

Fig. 7.1 – *The Attitude Control System*

tem stabilization, namely, the Attitude Control System (ACS) that controls the satellite attitude in the presence of disturbances by pointing the axes of

the spacecraft to the directions required for its mission (c.f. Figure 7.1). To do this, the satellite attitude is determined using measurements incoming from sensors and appropriate control torques that are exerted by actuators. The reaction wheels make part of the inertial actuators. The principle of the wheels is to create torques by accelerating or decelerating the rotor, which produces a reaction torque directly applied to the platform. In this thesis, our diagnosis approach is tested on a MATLAB/SIMULINK simulator (Figure 7.2) of a satellite Attitude Control System whose actuators are four reaction wheels. This case study was provided by Thales Alenia Space, France.

7.1 Presentation of the case study

7.1.1 Model description

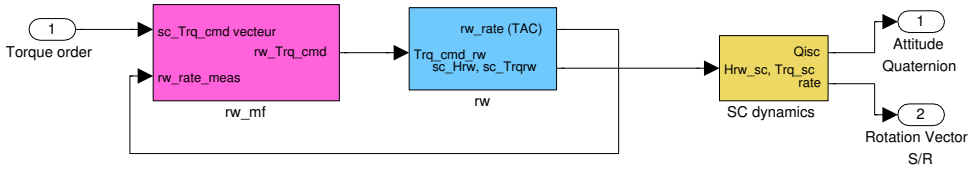

Fig. 7.2 – *The MATLAB/SIMULINK simulator of the ACS with reaction wheels*

The MATLAB/SIMULINK simulator is composed by three boxes :

- the "*rw_mf*" box models the system controller that computes the torque distribution on reaction wheels. This box takes as input the torque setpoint to be exerted on the spacecraft, the rotation velocity of every wheel (closed loop) and returns as output the torque setpoint for every reaction wheel.
- the "*rw*" box is the model of the reaction wheels. It takes as input the torque setpoint on every wheel. This setpoint is multiplied by the scale factor of the motor to obtain the actual torque that will be exerted by every wheel. Then, by subtracting viscous and Coulomb friction and by multiplying by an appropriate matrix that models the spatial configuration of wheel axes, the torque exerted by whole the system on the spacecraft is obtained. The kinetic momentum of the four reaction wheels is also obtained and returned as output of the "*rw*" box by applying the kinetic momentum theorem on the wheels system.
- the "*sc_dynamics*" box models the dynamics of the spacecraft. It takes as input the equivalent kinetic momentum (of the four wheels), the equivalent torque exerted by the system rotors linked with reaction wheels. Applying the kinetic momentum theorem to the spacecraft and changing reference frame, the rotation vector of the spacecraft is obtained. The attitude quaternion is then obtained from the rotation vector doing some mathematical operations.

The system observable variables that are available for diagnosis are the rotation vector measured by the gyroscope and the attitude quaternion measured by the star tracker system. The rotation vector can be obtained from the quaternion, this physical redundancy may be used to diagnose sensor faults. For actuator faults detection, we use the rotation vector measure.

7.1.2 The spacecraft equations

To get the equations of the spacecraft dynamics, we apply the kinetic momentum theorem to the system {spacecraft+wheels} in the inertial frame reference R :

$$\frac{d}{dt}_{/R}[\vec{H}_G] = \frac{d}{dt}_{/R}[\vec{H}_{GSpacecraft} + \vec{H}_{GWheels}] = \vec{M}_{GF_{ext}} \tag{7.1}$$

where :

- $\vec{H}_G$ is the kinetic momentum at G which, is the gravity center of the system
- $\vec{H}_{GSpacecraft}$ is the kinetic momentum of the spacecraft
- $\vec{H}_{GWheels}$ is the kinetic momentum of the reaction wheels
- $\vec{M}_{GF_{ext}}$ is the momentum of external forces exerted on the system. Here, it is the equivalent torque applied by rotors linked with reaction wheels denoted $\vec{T}_{rq_{sc}}$.

Let S denote the spacecraft reference frame (the reference frame linked to the spacecraft) and $\vec{\Omega}_{S/R}$ the rotation vector of S compared to R.
The kinetic momentum of the spacecraft is $\vec{H}_{GSpacecraft} = I_G.\vec{\Omega}_{S/R}$ where I_G is the inertia of the spacecraft.
By changing the reference frame from R to S we obtain :

$$\frac{d}{dt}_{/R}[\vec{H}_{GSpacecraft}] = \frac{d}{dt}_{/S}[\vec{H}_{GSpacecraft}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{GSpacecraft}$$

and

$$\frac{d}{dt}_{/R}[\vec{H}_{GWheels}] = \frac{d}{dt}_{/S}[\vec{H}_{GWheels}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{Wheels}$$

Therefore, the theorem of kinetic momuntum becomes :

$$\frac{d}{dt}_{/S}[\vec{H}_{GSpacecraft}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{GSpacecraft} + \frac{d}{dt}_{/S}[\vec{H}_{GWheels}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{Wheels} = \vec{T}_{rq_{sc}} \Leftrightarrow$$

$$\frac{d}{dt}_{/S}[I_G.\vec{\Omega}_{S/R}] + \vec{\Omega}_{S/R} \wedge I_G.\vec{\Omega}_{S/R} = \vec{T}_{rq_{sc}} - (\frac{d}{dt}_{/S}[\vec{H}_{GWheels}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{Wheels}) \Leftrightarrow$$

$$\frac{d}{dt}_{/S}[I_G.\vec{\Omega}_{S/R}] + \vec{\Omega}_{S/R} \wedge I_G.\vec{\Omega}_{S/R} = \vec{T}_{rq_{sc}} - \vec{H}_{rw_{sc}} \tag{7.2}$$

where :

- $\vec{H}_{rw_{sc}} = \frac{d}{dt}_{/S}[\vec{H}_{GWheels}] + \vec{\Omega}_{S/R} \wedge \vec{H}_{Wheels}$ represents the torque exchanged between the reaction wheels and the spacecraft.
- $\frac{d}{dt}_{/S}[\vec{H}_{GWheels}]$ represents the torque applied to the spacecraft due to the velocity variation of wheels. The wheel velocity must continuously increase to compensate a torque with non null average value. Since the wheel velocity is limited by the maximal velocity of its motor, wheels have to be kept out of the saturation limite.
- $\vec{\Omega}_{S/R} \wedge \vec{H}_{Wheels}$ represents the gyroscopic coupling due to the presence of wheels. This torque is low in the case of reaction wheels.

Let $\vec{\Omega}_{S/R} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$, $\vec{T}_{rq_{sc}} = \begin{pmatrix} T_{rq_{sc1}} \\ T_{rq_{sc2}} \\ T_{rq_{sc3}} \end{pmatrix}$ and $\vec{H}_{rw_{sc}} = \begin{pmatrix} H_{rw_{sc1}} \\ H_{rw_{sc2}} \\ H_{rw_{sc3}} \end{pmatrix}$
The spacecraft equation becomes :

$$\boxed{I_G.\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} + \begin{pmatrix} p \\ q \\ r \end{pmatrix} \wedge I_G.\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} T_{rq_{sc_1}} - H_{rw_{sc1}} \\ T_{rq_{sc_2}} - H_{rw_{sc2}} \\ T_{rq_{sc_3}} - H_{rw_{sc3}} \end{pmatrix}} \tag{7.3}$$

Numerically, in our case study the spacecraft inertia is $I_G = \begin{pmatrix} 600 & 0 & 0 \\ 0 & 700 & 0 \\ 0 & 0 & 600 \end{pmatrix}$
Developing Equation 7.3, we get :

$$\begin{cases} \dot{p} &= \frac{1}{6} q.r + \frac{1}{600}(T_{rq_{sc1}} - H_{rw_{sc1}}) \\ \dot{q} &= \frac{1}{700}(T_{rq_{sc2}} - H_{rw_{sc2}}) \\ \dot{r} &= \frac{-1}{6} p.q + \frac{1}{600}(T_{rq_{sc3}} - H_{rw_{sc3}}) \end{cases} \tag{7.4}$$

The polynomial non linearity present in Equation 7.4 is modeled in a non linear space state representation as follows :

$$\begin{cases} \dot{x} = f(x,u) \\ y = g(x,u) \end{cases} \tag{7.5}$$

with $u = [H_{rw_{sc1}}, H_{rw_{sc2}}, H_{rw_{sc3}}, T_{rq_{sc1}}, T_{rq_{sc2}}, T_{rq_{sc3}}]$, $x = [p,r,q]^T$ and $y = x$ (i.e. $g(x,u) = x$). Therefore, residuals generation approach for non linear systems is required. In a general case, an extension of the parity space approach to polynomial systems can be applied (c.f. Staroswiecki and Comtet-Varga (2001a)). Here, since $y = x$, it is a degenerated case. Analytic redundancy relations are directly given by the state evolution equation. Notice, that $H_{rw_{sci}}$ and $H_{rw_{sci}}$, $i = 1..3$ are not measured. They will be expressed in terms of observable variables by trickily connecting the spacecraft and actuator equations in Section 7.1.3.

7.1.3 Actuator (reaction wheels) equations

Let us define the following variables :

- T_{rw_i} is the torque exerted by the reaction wheel i
- T_{rot_i} is the torque generated by the rotor associated to the wheel i
- T_{cmd_i} is the torque setpoint for the reaction wheel i
- T_{fric_i} is the viscous and the Coulomb friction torque exerted on the wheel i
- H_{rw_i} is the kinetic momentum of the wheel i
- α_{m_i} is the factor scale of the associated motor of the wheel i, when no fault $\alpha_{m_1} = \alpha_{m_2} = \alpha_{m_3} = \alpha_{m_4} = \alpha_m = 1.01$
- i is the index of the wheel, $i = 1..4$

The torque exerted by a reaction wheel i is equal to the difference between the torque exerted by the associated rotor and the friction (Coulomb and viscous) torque.

$$T_{rw_i} = T_{rot_i} - T_{fric_i} \tag{7.6}$$

The torque exerted by the rotor of the wheel i is given as follows :

$$T_{rot_i} = \alpha_{m_i} \times T_{cmd_i} \tag{7.7}$$

Now let us take into account the saturation phenomenon. The maximal torque that can be provided by the motor associated to the wheel i is $T_{rot_{max}} = 0.15\ N.m$.

$$\boxed{T_{rot_i} = \begin{cases} \alpha_{m_i} \times T_{cmd_i} & \text{if } |T_{cmd_i}| \leq \frac{T_{rot_{max}}}{\alpha_{m_i}} \\ sign(T_{cmd_i}).T_{rot_{max}} & \text{otherwise} \end{cases}} \tag{7.8}$$

The actual torque exerted on the spacecraft by the rotation of all reaction wheels is :

$$\boxed{\vec{T}_{rq_{sc}} = C_{as}^{T} \begin{pmatrix} T_{rw1} \\ T_{rw2} \\ T_{rw3} \\ T_{rw4} \end{pmatrix}} \tag{7.9}$$

Where $C_{as} = \begin{pmatrix} 0.7071 & 0.5000 & 0.5000 \\ 0.7071 & 0.5000 & -0.5000 \\ 0.7071 & -0.5000 & -0.5000 \\ 0.7071 & -0.5000 & 0.5000 \end{pmatrix}$ models the configuration of the spin axes of the wheels in the space.
The theorem of kinetic momentum applied to the reaction wheel i gives :

$$\frac{dH_{rw_i}}{dt} = -T_{rw_i} \tag{7.10}$$

Equation 7.10 means that the derivative of the kinetic momentum of the wheel i is equal to the torque exerted on the wheel. Since T_{rw_i} is the torque is the torque exerted by the wheel i on the spacecraft, $-T_{rw_i}$ is the torque exerted by the spacecraft on the wheel i.
Now, let us take into account the saturation phenomenon on the kinetic momentum. The maximal kinetic momentum of a wheel i is $H_{rw_{max}} = 18\ Kg.m^2.s^{-1}$. It models the maximal velocity of the wheel. Consequently :

$$\boxed{H_{rw_i} = \begin{cases} -\int T_{rw_i} & \text{if } |-\int T_{rw_i}| \leq H_{rw_{max}} \\ sign(-\int T_{rw_i}).H_{rw_{max}} & \text{otherwise} \end{cases}} \tag{7.11}$$

The kinetic momentum of the four wheels $\vec{H}_{rw_{sc}}$ exchanged with the spacecraft is given by :

$$\boxed{\vec{H}_{rw_{sc}} = C_{as}^{T} \begin{pmatrix} H_{rw_1} \\ H_{rw_2} \\ H_{rw_3} \\ H_{rw_4} \end{pmatrix}} \tag{7.12}$$

Friction equations

For a wheel $i, i = 1..4$ the friction torque is given by :

$$T_{fric_i} = T_{viscous\_fric_i} + T_{Coulomb\_fric_i} \tag{7.13}$$

where :

$$T_{viscous\_fric_i} = f_{viscous}.\dot{\theta}_i \tag{7.14}$$

$$T_{Coulomb\_fric_i} = sign(\dot{\theta}_i).f_{Coulomb} \tag{7.15}$$

Where :

- $f_{viscous}$ is the viscous friction coefficient, numerically $f_{viscous} = 0.27\ 10^{-4}\ N.m.s$
- $f_{Coulomb}$ is the Coulomb friction coefficient, numerically $f_{Coulomb} = 0.005\ N.m$
- $\dot{\theta}_i = \frac{H_{rw_i}}{I}$ denotes the rotation velocity of wheel i

– I is the inertia of wheel i, numerically $I = 0.1322\ Kg.m^2$ (the four wheels have the same inertia)

Hence, from 7.6 the torque exerted by a reaction wheel i is given by :

$$\boxed{T_{rw_i} = T_{rot_i} - f_{viscous}\frac{H_{rw_i}}{I} - sign(\dot{H}_{rw_i})\ f_{Coulomb}} \tag{7.16}$$

Non linearity modeling

The non linearities present in the model of wheels are saturations (Equations 7.8 and 7.11) and Coulomb friction (Equation 7.15). We have chosen to model them with Piece-Wise-Affine (PWA) functions. In our diagnosis model, each linear region is modeled by an additional operating mode.

7.2 The specification of the diagnosis problem

The diagnosis problem focus on actuator-faults. The attitude control of the spacecraft is achieved by means of inertial actuators which are the four reaction wheels. For each wheel i, F_i models a failure of the associated motor i.e. the wheel does not rotate (the associated factor scale is $\alpha_{m_i} = 0$).

7.3 Problem formalization in the hybrid modeling framework

The system is composed by 5 components : the spacecraft itself and the four reaction wheels. Each component is modeled as a hybrid automaton. The behavior of whole the system is seen as the synchronous product of these component automata.

7.3.1 The spacecraft hybrid model

The spacecraft is modeled as a hybrid automaton $A(S)$ that contains only one mode modeling the nominal behavior described by Equations 7.4. This component is assumed to be fault-free.

7.3.2 The reaction wheel hybrid model

Each reaction wheel is modeled as a hybrid automaton. Two types of modes are distinguished : nominal and faulty modes. The underlying behavior of each mode is described by linear equations.

Model of the nominal behavior

For each reaction wheel i, each linear region is modeled as a different configuration, leading to an operating mode at the level of the whole system.

– The non linearity in Equation 7.8 is modeled by 3 configurations as shown in Table 7.1.
– The non linearity in Equation 7.11 is modeled by 3 configurations as shown in Table 7.2.

Configuration	Equation	Condition
a_{i_1}	$T_{rot_i} = \alpha_{m_i} \times T_{cmd_i}$	$\lvert T_{cmd_i} \rvert \leq \frac{T_{rot_{max}}}{\alpha_{m_i}}$
a_{i_2}	$T_{rot_i} = T_{rot_{max}}$	$T_{cmd_i} > \frac{T_{rot_{max}}}{\alpha_{m_i}}$
a_{i_3}	$T_{rot_i} = -T_{rot_{max}}$	$T_{cmd_i} < -\frac{T_{rot_{max}}}{\alpha_{m_i}}$

TAB. 7.1 – *System configurations that model the torque saturation*

Configuration	Equation	Condition
b_{i_1}	$H_{rw_i} = -\int T_{rw_i}$	$\lvert -\int T_{rw_i} \rvert \leq H_{rw_{max}}$
b_{i_2}	$H_{rw_i} = H_{rw_{max}}$	$-\int T_{rw_i} > H_{rw_{max}}$
b_{i_3}	$H_{rw_i} = -H_{rw_{max}}$	$-\int T_{rw_i} < H_{rw_{max}}$

TAB. 7.2 – *System configurations that model the momentum saturation*

– The non linearity in Equation 7.15 is modeled by 2 configurations as shown in Table 7.3.

Configuration	Equation	Condition
c_{i_1}	$T_{Coulomb\_fric_i} = f_{Coulomb}$	$sign(H_{rw_i}) > 0$
c_{i_2}	$T_{Coulomb\_fric_i} = -f_{Coulomb}$	$sign(H_{rw_i}) < 0$

TAB. 7.3 – *System configurations that model the Coulomb friction*

As an example, when the wheel i is in configuration $a_{i_1} b_{i_1} c_{i_1}$ Equations 7.16 and 7.11 become :

$$T_{rw_i} = \alpha_{m_i} \times T_{cmd_i} - f_{viscous} \frac{H_{rw_i}}{I} - f_{Coulomb} \tag{7.17}$$

$$H_{rw_i} = -\int T_{rw_i} \tag{7.18}$$

respectively.
The number of configurations is : $3 \times 3 \times 2 = 18$ that model the regions of linear behavior of the system. A wheel $i, i = 1..4$ is in configuration : $a_{i_k} b_{i_l} c_{i_m}$, $k = 1..3$, $l = 1..3$ and $m = 1..2$. However, some configurations are impossible because of physical considerations. Since $H_{rw_i} = I.\theta_i$, $b_{i_2} \Rightarrow c_{i_1}$ and $b_{i_3} \Rightarrow c_{i_2}$. Then the number of nominal modes is $3 \times 2 \times 2 = 12$. The configuration conditions shown in Tables 7.1, 7.2 and 7.3 define the nominal transition guards.
The hybrid automaton $A_N(w_i)$ representing the nominal behavior of a wheel i is given in Figure 7.3.

Model of the faulty behavior

We focus on actuator faults (F_1, F_2, F_3 and F_4) as described in Section 7.2. We associate an unobservable fault event f_j to each fault F_j that models the transition between the nominal mode and the faulty mode. For sake of simplicity, we only consider single faults (i.e. we do not consider the case of two consecutive fault events). A fault event f_j may occur in any nominal mode q_{N_i}, $i = 1..12^4$ and leads to a faulty mode $q_{N_i F_j}$. The faulty behavior of a wheel i is described by the hybrid automaton shown in Figure 7.4.

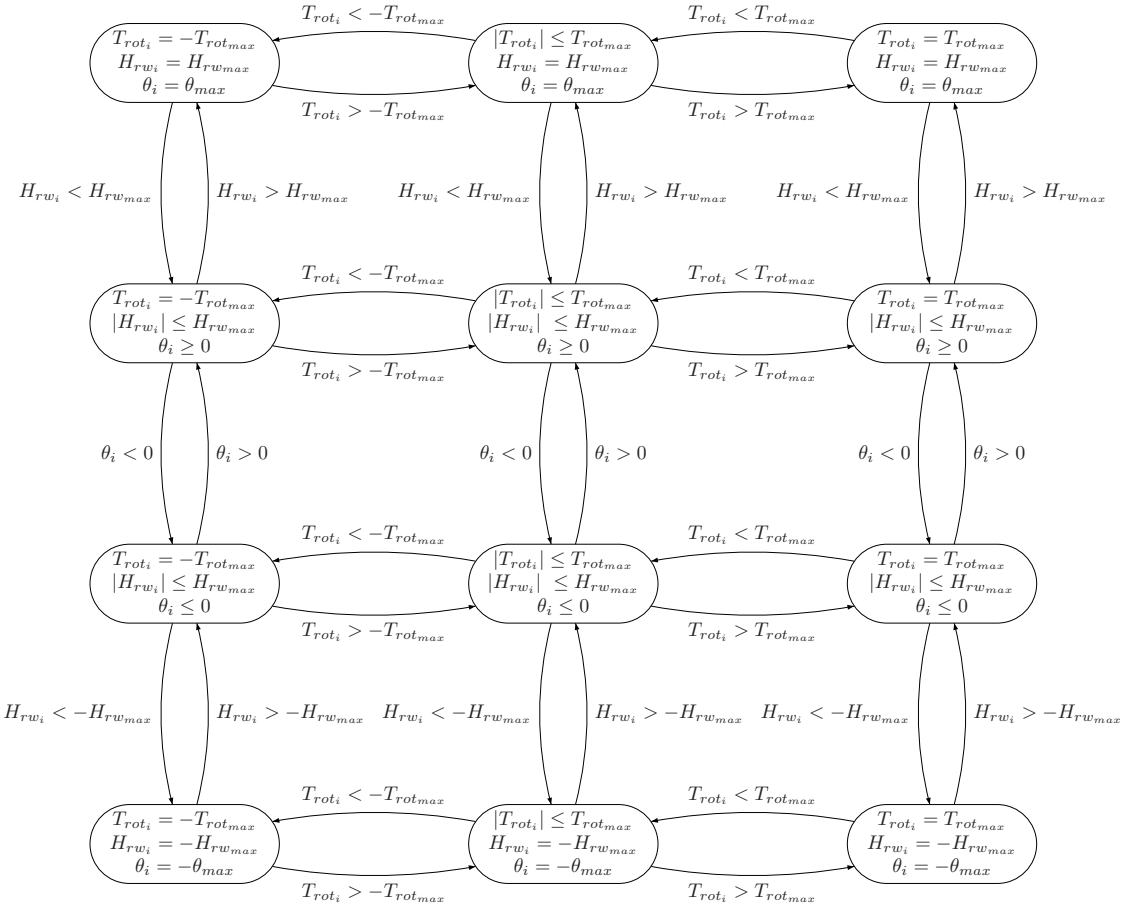

FIG. 7.3 – *The hybrid automaton $A_N(w_i)$ that models the nominal behavior of wheel i*

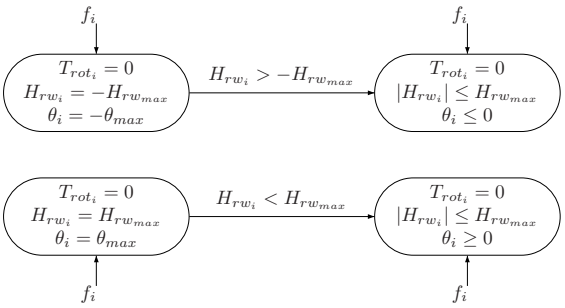

FIG. 7.4 – *The hybrid automaton $A_F(w_i)$ that models the faulty behavior of wheel i*

The model of nominal and faulty behavior of a wheel

A wheel is composed by a flywheel that exchanges the kinetic momentum H_{rw_i} with the spacecraft and a motor that exerts the motor torque T_{rot_i}. The automata models of the flywheel and motor of wheel *i* are provided in Figures 7.5 and 7.6 respectively. The whole automaton model of the wheel *i* including nominal and fault modes can be obtained by composing the flywheel and the motor automata.

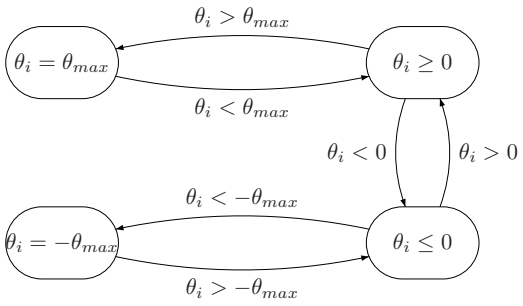

FIG. 7.5 – *The hybrid automaton model of the flywheel associated to wheel i*

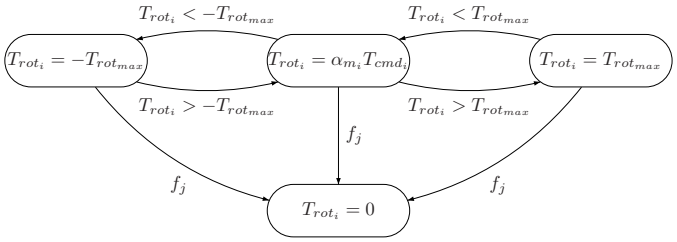

FIG. 7.6 – *The hybrid automaton model of the motor associated to wheel i*

The diagnosis model of the system

The behavior of each wheel (both nominal and faulty) is modeled by a hybrid automaton $A(w_i)$ that contains $4 + 12 = 16$ operating modes and obtained by combining $A_N(w_i)$ and $A_F(w_i)$. $A(w_i)$ can also obtained by composing flywheel and motor automata : 16 = 4×4.
The hybrid automaton of the whole system is $A(System) = A(w_1)||A(w_2)||A(w_3)||A(w_4)||A(S)$ that contains $16^4 \times 1 = 65536$ modes. Since we assume the single fault assumption, modes that model multiple faults are deleted. Then the number of all system modes is reduced to : $12^4 + 12^3 \times 4 \times 4 = 48384$ modes.

7.4 Diagnosis scheme

7.4.1 Diagnosis of the underlying multimode system

Generation of residuals

First, we assume that the torque setpoint T_{cmd_i} for each reaction wheel i is measured i.e. that the controller ("rw_mf") output vector $[T_{cmd_1}, T_{cmd_2}, T_{cmd_3}, T_{cmd_4}]^T$ (c.f. Section 7.1.1) is measured.
The connection between the spacecraft and the actuator equations (Equations 7.4 and 7.16, respectively) is achieved by means of equations 7.9 and 7.12.
Analytic redundancy relations for each system mode (nominal and faulty) are obtained by merging spacecraft and actuator equations. Non observable variables (connection variables) are eliminated by manipulating spacecraft and actuator equations.
By simulating the system w.r.t the scenario provided by THALES ALENIA SPACE, we notice that it starts in the nominal mode q_{N_1} corresponding to the wheel configurations : $a_{11}b_{11}c_{11}.a_{21}b_{21}c_{22}.a_{32}b_{31}c_{31}.a_{41}b_{41}c_{42}$. In this mode, the analytic redundancy relations are generated as follows :
Equations 7.16 becomes :

$$T_{rw_i} = \alpha_{m_i} \times T_{cmd_i} - f_{viscous}\frac{H_{rw_i}}{I} - f_{Coulomb}, \text{ for } i = 1 \tag{7.19}$$

$$T_{rw_i} = T_{rot_{max}} - f_{viscous}\frac{H_{rw_i}}{I} - f_{Coulomb}, \text{ for } i = 3 \tag{7.20}$$

$$T_{rw_i} = \alpha_{m_i} \times T_{cmd_i} - f_{viscous}\frac{H_{rw_i}}{I} + f_{Coulomb}, \text{ for } i = 2 \text{ and } i = 4 \tag{7.21}$$

and Equation 7.11 becomes :

$$H_{rw_i} = -\int T_{rw_i}, \text{ for } i = 1..4 \tag{7.22}$$

We proceed as follows
Actuator Equations 7.19 (7.20, 7.21) and 7.22 give us :

- for $i = 1$:

$$\begin{cases} \dot{H}_{rw_i} = -\frac{f_{viscous}}{I}H_{rw_i} + (\alpha_{m_i}\ T_{cmd_i} - f_{Coulomb}) \\ T_{rw_i} = -\dot{H}_{rw_i} \end{cases} \tag{7.23}$$

- for $i = 3$:

$$\begin{cases} \dot{H}_{rw_i} = -\frac{f_{viscous}}{I}H_{rw_i} + (T_{rot_{max}} - f_{Coulomb}) \\ T_{rw_i} = -\dot{H}_{rw_i} \end{cases} \tag{7.24}$$

- for $i = 2$ and $i = 4$:

$$\begin{cases} \dot{H}_{rw_i} = -\frac{f_{viscous}}{I}H_{rw_i} + (\alpha_{m_i}\ T_{cmd_i} + f_{Coulomb}) \\ T_{rw_i} = -\dot{H}_{rw_i} \end{cases} \tag{7.25}$$

Consequently :

- for $i = 1$:

$$\begin{cases} \dot{H}_{rw_i} + \frac{f_{viscous}}{I}H_{rw_i} = \alpha_{m_i}\ T_{cmd_i} - f_{Coulomb} \\ \dot{T}_{rw_i} + \frac{f_{viscous}}{I}T_{rw_i} = -\alpha_{m_i}\ \dot{T}_{cmd_i} \end{cases} \tag{7.26}$$

– for $i = 3$:

$$\begin{cases} \dot{H}_{rw_i} + \frac{f_{viscous}}{I} H_{rw_i} = T_{rot_{max}} - f_{Coulomb} \\ \dot{T}_{rw_i} + \frac{f_{viscous}}{I} T_{rw_i} = 0 \end{cases} \tag{7.27}$$

– for $i = 2$ and $i = 4$:

$$\begin{cases} \dot{H}_{rw_i} + \frac{f_{viscous}}{I} H_{rw_i} = \alpha_{m_i}\ T_{cmd_i} + f_{Coulomb} \\ \dot{T}_{rw_i} + \frac{f_{viscous}}{I} T_{rw_i} = -\alpha_{m_i}\ \dot{T}_{cmd_i} \end{cases} \tag{7.28}$$

Then, by replacing Equations 7.9 and 7.12 in Equation 7.4, we obtain :

$$\begin{cases} \dot{p} = \frac{1}{6} q.r + \frac{1}{600}[0.7071(T_{rw_1} - H_{rw_1}) + 0.7071(T_{rw_2} - H_{rw_2}) + \\ \qquad 0.7071(T_{rw_3} - H_{rw_3}) + 0.7071(T_{rw_4} - H_{rw_4})] \\ \dot{q} = \frac{1}{700}[0.5(T_{rw_1} - H_{rw_1}) + 0.5(T_{rw_2} - H_{rw_2}) - \\ \qquad 0.5(T_{rw_3} - H_{rw_3}) - 0.5(T_{rw_4} - H_{rw_4})] \\ \dot{r} = \frac{-1}{6} p.q + \frac{1}{600}[0.5(T_{rw_1} - H_{rw_1}) - 0.5(T_{rw_2} - H_{rw_2}) - \\ \qquad 0.5(T_{rw_3} - H_{rw_3}) + 0.5(T_{rw_4} - H_{rw_4})] \end{cases} \tag{7.29}$$

By derivating Equation 7.29, we obtain :

$$\begin{cases} \ddot{p} = \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{1}{600}[0.7071(\dot{T}_{rw_1} - \dot{H}_{rw_1}) + 0.7071(\dot{T}_{rw_2} - \dot{H}_{rw_2}) + \\ \qquad 0.7071(\dot{T}_{rw_3} - \dot{H}_{rw_3}) + 0.7071(\dot{T}_{rw_4} - \dot{H}_{rw_4})] \\ \ddot{q} = \frac{1}{700}[0.5(\dot{T}_{rw_1} - \dot{H}_{rw_1}) + 0.5(\dot{T}_{rw_2} - \dot{H}_{rw_2}) - \\ \qquad 0.5(\dot{T}_{rw_3} - \dot{H}_{rw_3}) - 0.5(\dot{T}_{rw_4} - \dot{H}_{rw_4})] \\ \ddot{r} = \frac{-1}{6}(p.\dot{q} + \dot{p}.q) + \frac{1}{600}[0.5(\dot{T}_{rw_1} - \dot{H}_{rw_1}) - 0.5(\dot{T}_{rw_2} - \dot{H}_{rw_2}) - \\ \qquad 0.5(\dot{T}_{rw_3} - \dot{H}_{rw_3}) + 0.5(\dot{T}_{rw_4} - \dot{H}_{rw_4})] \end{cases} \tag{7.30}$$

Finally, $\frac{f_{viscous}}{I} \times 7.29 + 7.30 \Leftrightarrow$

$$\begin{cases} \frac{f_{viscous}}{I}\dot{p} + \ddot{p} = \frac{f_{viscous}}{6.I} q.r + \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{1}{600}[0.7071(-\alpha_{m_i}\ \dot{T}_{cmd_1} - \alpha_{m_i}\ T_{cmd_1} \\ -f_{Coulomb}) + 0.7071(-\alpha_{m_i}\ \dot{T}_{cmd_2} - \alpha_{m_i}\ T_{cmd_2} + f_{Coulomb}) + 0.7071(-T_{rot_{max}} \\ -f_{Coulomb}) + 0.7071(-\alpha_{m_i}\ \dot{T}_{cmd_4} - \alpha_{m_i}\ T_{cmd_4} + f_{Coulomb})] \\ \\ \frac{f_{viscous}}{I}\dot{q} + \ddot{q} = \frac{1}{700}[0.5(-\alpha_{m_i}\ \dot{T}_{cmd_1} - \alpha_{m_i}\ T_{cmd_1} - f_{Coulomb}) + 0.5(-\alpha_{m_i}\ \dot{T}_{cmd_2} \\ -\alpha_{m_i}\ T_{cmd_2} + f_{Coulomb}) - 0.5(-T_{rot_{max}} - f_{Coulomb}) - 0.5(-\alpha_{m_i}\ \dot{T}_{cmd_4} \\ -\alpha_{m_i}\ T_{cmd_4} + f_{Coulomb}))] \\ \\ \frac{f_{viscous}}{I}\dot{r} + \ddot{r} = \frac{-f_{viscous}}{6.I} p.q + \frac{-1}{6}(p.\dot{q} + \dot{p}.q) + \frac{1}{600}[0.5(-\alpha_{m_i}\ \dot{T}_{cmd_1} - \alpha_{m_i}\ T_{cmd_1} \\ -f_{Coulomb}) - 0.5(-\alpha_{m_i}\ \dot{T}_{cmd_2} - \alpha_{m_i}\ T_{cmd_2} + f_{Coulomb}) - 0.5(-T_{rot_{max}} \\ -f_{Coulomb}) + 0.5(-\alpha_{m_i}\ \dot{T}_{cmd_4} - \alpha_{m_i}\ T_{cmd_4} + f_{Coulomb})] \end{cases} \tag{7.31}$$

The analytic redundancy relations of mode q_{N_1} (and their associated residuals) are then given as follows :

$$\begin{cases} ARR_{N11} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_1} + \dot{T}_{cmd_2} \\ +\dot{T}_{cmd_4}) + (T_{cmd_1} + T_{cmd_2} + \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N11}) \\ \\ ARR_{N12} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_1} + \dot{T}_{cmd_2} - \dot{T}_{cmd_4}) + (T_{cmd_1} + T_{cmd_2} - \\ \frac{T_{rot_{max}}}{\alpha_m} - T_{cmd_4})] = 0 \ (r_{N12}) \\ \\ ARR3_{N13} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(\dot{T}_{cmd_1} - \dot{T}_{cmd_2} \\ +\dot{T}_{cmd_4}) + (T_{cmd_1} - T_{cmd_2} - \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N13}) \end{cases} \tag{7.32}$$

The analytic redundancy relations in each mode are different. Hence, the mode signatures are different, as a consequence the hybrid system is diagnosable w.r.t the CS sufficient condition (Theorem 5.2.2, Chapter 5) by means of Corollary 5.1. In this case the diagnoser of the hybrid system is simply obtained from the behavior automaton and contains 48348 states.

Towards an on-line analytic redundancy relation generation

To optimize the diagnosis scheme we avoid to build the automaton of the hybrid system (the synchronous product). We keep the non linearities obtaining a generic analytic redundancy relations that contain saturation functions. From a current mode we compute the possible destination modes by considering component automata. In the worst case, the number of possible destination modes is : $4 \times 5 = 20$ modes (under the hypothesis of asynchronous guard transition validation). For all these possible modes we instantiate the generic analytic redundancy relations by replacing the saturation functions by their corresponding instances w.r.t configurations shown in Tables 7.1, 7.2 and 7.3. Hence, in the worst case, we instantiate on-line $3 \times 20 = 60$ analytic redundancy relations. By checking on-line the consistency of associated residuals we detect the mode change (this is guaranteed by the diagnosability property of the system). Hence, the system mode can be tracked on-line.

7.5 Simulation and results

The diagnosis module of the system is implemented in Matlab/Simulink following the diagnosis scheme presented in Figure 4.7 of Chapter 4.

- the residual bench : let $q_{N_1F_1}$, $q_{N_1F_2}$, $q_{N_1F_3}$ and $q_{N_1F_4}$ model the fault modes after the occurrence of fault events f_1, f_2, f_3 and f_4 in the mode q_{N_1}. The analytic redundancy relations of faulty modes $q_{N_1F_1}$ and $q_{N_1F_2}$, $q_{N_1F_3}$ and $q_{N_1F_4}$ are computed as explained in Section 7.4.1 and provided by Equations 7.33, 7.34, 7.35 and 7.36, respectively. The

sampling period is $T_s = 10^{-4}$ s.

$$\begin{cases} ARR_{N1F11} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_2} \\ +\dot{T}_{cmd_4}) + (T_{cmd_2} + \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N1F11}) \\ \\ ARR_{N1F12} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_2} - \dot{T}_{cmd_4}) + (T_{cmd_2} - \\ \frac{T_{rot_{max}}}{\alpha_m} - T_{cmd_4})] = 0 \ (r_{N1F12}) \\ \\ ARR_{N1F13} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(-\dot{T}_{cmd_2} \\ +\dot{T}_{cmd_4}) + (-T_{cmd_2} - \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N1F13}) \end{cases} \tag{7.33}$$

$$\begin{cases} ARR_{N1F21} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_1} \\ +\dot{T}_{cmd_4}) + (T_{cmd_1} + \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N1F21}) \\ \\ ARR_{N1F22} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_1} - \dot{T}_{cmd_4}) + (T_{cmd_1} - \\ \frac{T_{rot_{max}}}{\alpha_m} - T_{cmd_4})] = 0 \ (r_{N1F22}) \\ \\ ARR_{N1F23} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(\dot{T}_{cmd_1} \\ +\dot{T}_{cmd_4}) + (T_{cmd_1} - \frac{T_{rot_{max}}}{\alpha_m} + T_{cmd_4})] = 0 \ (r_{N1F23}) \end{cases} \tag{7.34}$$

$$\begin{cases} ARR_{N1F31} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_1} \\ +\dot{T}_{cmd_2} + \dot{T}_{cmd_4}) + (T_{cmd_1} + T_{cmd_2} + T_{cmd_4})] = 0 \ (r_{N1F31}) \\ \\ ARRr_{N1F32} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_1} + \dot{T}_{cmd_2} - \dot{T}_{cmd_4}) + (T_{cmd_1} \\ +T_{cmd_2} - T_{cmd_4})] = 0 \ (r_{N1F32}) \\ \\ ARRr_{N1F33} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(\dot{T}_{cmd_1} \\ -\dot{T}_{cmd_2} + \dot{T}_{cmd_4}) + (T_{cmd_1} - T_{cmd_2} + T_{cmd_4})] = 0 \ (r_{N1F33}) \end{cases} \tag{7.35}$$

$$\begin{cases} ARR_{N1F41} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_1} \\ +\dot{T}_{cmd_2}) + (T_{cmd_1} + T_{cmd_2} + \frac{T_{rot_{max}}}{\alpha_m})] = 0 \ (r_{N1F41}) \\ \\ ARR_{N1F42} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_1} + \dot{T}_{cmd_2}) + (T_{cmd_1} + T_{cmd_2} \\ -\frac{T_{rot_{max}}}{\alpha_m}] = 0 \ (r_{N1F42}) \\ \\ ARR_{N1F43} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(\dot{T}_{cmd_1} \\ -\dot{T}_{cmd_2}) + (T_{cmd_1} - T_{cmd_2} - \frac{T_{rot_{max}}}{\alpha_m})] = 0 \ (r_{N1F43}) \end{cases} \tag{7.36}$$

Let q_{N_2} denote the nominal mode that corresponds to the wheel configurations $a_{11}b_{11}c_{11}.a_{21}b_{21}c_{22}.a_{31}b_{31}c_{31}.a_{41}b_{41}c_{42}$ and $q_{N_2F_1}$ the corresponding destination mode after the occurrence of fault event f_1. These analytic redundancy relations are given by Equations 7.37 and

7.38.

$$\begin{cases} ARR_{N21} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[(\dot{T}_{cmd_1} \\ +\dot{T}_{cmd_2} + \dot{T}_{cmd_3} + \dot{T}_{cmd_4}) + (T_{cmd_1} + T_{cmd_2} + T_{cmd_3} + T_{cmd_4})] = 0 \\ (r_{N21}) \\ \\ ARR_{N22} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_1} + \dot{T}_{cmd_2} - \dot{T}_{cmd_3} - \dot{T}_{cmd_4}) \\ +(T_{cmd_1} + T_{cmd_2} - T_{cmd_3} - T_{cmd_4})] = 0 \\ (r_{N22}) \\ \\ ARR3_{N23} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(\dot{T}_{cmd_1} \\ -\dot{T}_{cmd_2} - \dot{T}_{cmd_3} + \dot{T}_{cmd_4}) + (T_{cmd_1} - T_{cmd_2} - T_{cmd_3} + T_{cmd_4})] = 0 \\ (r_{N23}) \end{cases} \tag{7.37}$$

$$\begin{cases} ARR_{N2F11} : \frac{f_{viscous}}{I}\dot{p} + \ddot{p} - \frac{f_{viscous}}{6.I}q.r - \frac{1}{6}(q.\dot{r} + \dot{q}.r) + \frac{0.7071\alpha_m}{600}[\dot{T}_{cmd_2} \\ +\dot{T}_{cmd_3} + \dot{T}_{cmd_4}) + (T_{cmd_2} + T_{cmd_3} + T_{cmd_4})] = 0 \ (r_{N2F11}) \\ \\ ARR_{N2F12} : \frac{f_{viscous}}{I}\dot{q} + \ddot{q} + \frac{0.5\alpha_m}{700}[(\dot{T}_{cmd_2} - \dot{T}_{cmd_3} - \dot{T}_{cmd_4}) + T_{cmd_2} \\ -T_{cmd_3} - T_{cmd_4})] = 0 \ (r_{N2F12}) \\ \\ ARR3_{N2F13} : \frac{f_{viscous}}{I}\dot{r} + \ddot{r} + \frac{f_{viscous}}{6.I}p.q + \frac{1}{6}(p.\dot{q} + \dot{p}.q) + \frac{0.5\alpha_m}{600}[(- \\ \dot{T}_{cmd_2} - \dot{T}_{cmd_3} + \dot{T}_{cmd_4}) + (-T_{cmd_2} - T_{cmd_3} + T_{cmd_4})] = 0 \ (r_{N2F13}) \end{cases} \tag{7.38}$$

- The residual filter : the residual computation involves variables of order magnitude 10^{-5}. The threshold is set as follows : $Threshold = 5.\ 10^{-10}$. To compute the residuals, we require the derivatives of p, q and r at orders 1 and 2. Hence, the filter sensitivity T_{Filter} must be higher than $3.T_s$. Furthermore, transitions to (or from) saturation modes are slow, hence we require time to detect them, T_{filter} is $0.01s$.
- the hybrid diagnoser : part of the diagnoser is implemented to run the scenarios presented below. The diagnoser returns the number of estimated system mode. The mapping between system modes and associated numbers can be found in Table 7.4.

Three scenarios are considered to illustrate our approach. Unfortunately, it was not possible to exhibit a scenario for active diagnosis due to the diagnosability of the considered system.

7.5.1 Scenario 1

The system starts in the nominal mode q_{N_1}. At $t = 14.05s$ the system mode changes autonomously from q_{N_1} to mode q_{N_2}.

$$q_{N_1} \xrightarrow{T_{rot_3} < 0.15 \text{ at } t=14.05s} q_{N_2}$$
Simulation time = $40s$

Figure 7.7 shows the evolution of non observable variables : the motor torque T_{rot_i}, the kinetic momentum H_{rw_i} and the rotation velocity θ_i, of each wheel i. The simulation duration is 40 s. We notice that the motor torque T_{rot_3} of wheel 3 is equal to $T_{rot_{max}} = 0.15\ N.m$ until $t = 14.05\ s$.

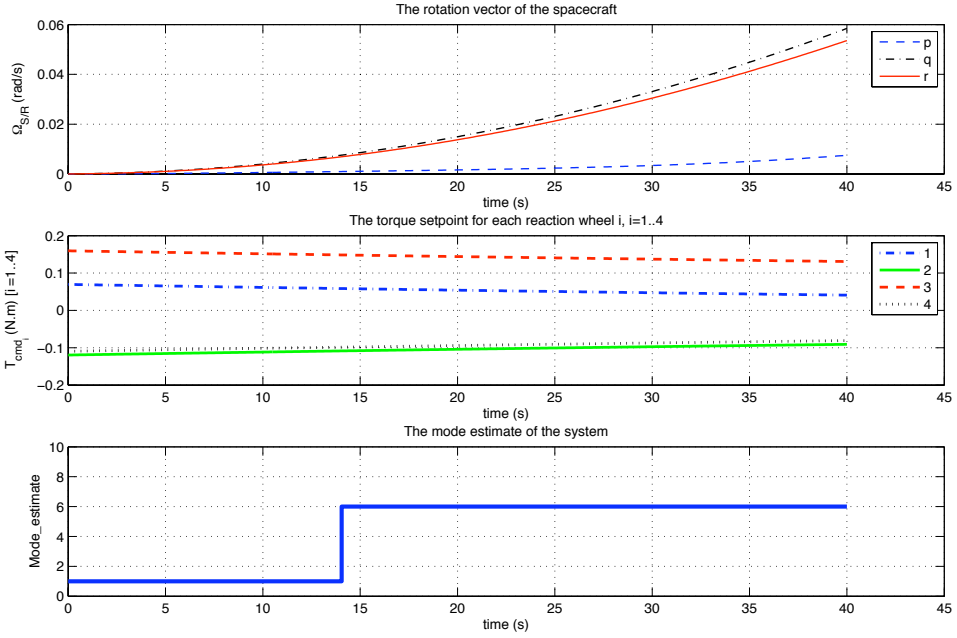

FIG. 7.7 – *Scenario 1 : non observables variables*

Then at $t = 14.05\ s$ T_{rot_3} leaves the saturation region.
The residuals of modes q_{N_1} and q_{N_2} are computed from the observable

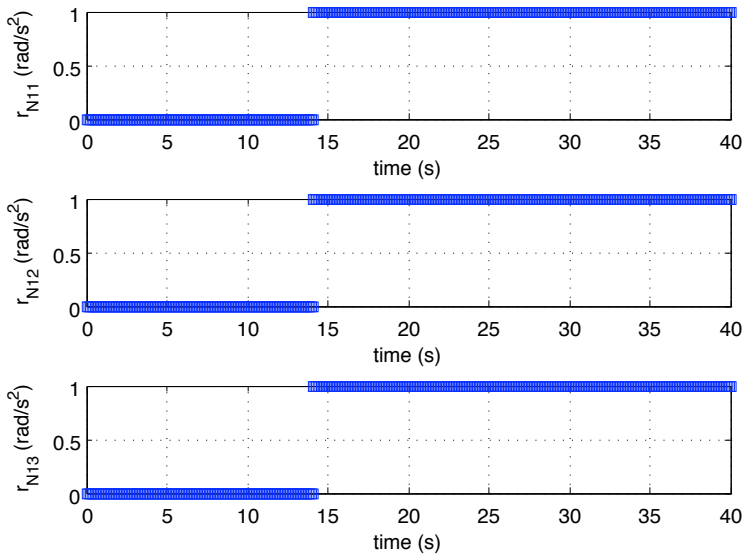

FIG. 7.8 – *Scenario 1 : residuals of mode* q_{N_1}

variables (and their derivatives) as shown in Figures 7.8 and 7.9, respectively. Figure 7.10 provides the mode estimate and the observable variables : $T_{cmd_i}, i = 1..4$ and $\vec{\Omega}_{S/R}$. Notice the mapping between residual values and the mode estimate. The mapping between system configuration and mode estimate can also be verified by comparing Figure 7.7 and Figure 7.10.
The time of the autonomous mode change is detected by the diagnosis module. Notice that the detection of mode transitions into flywheel satu-

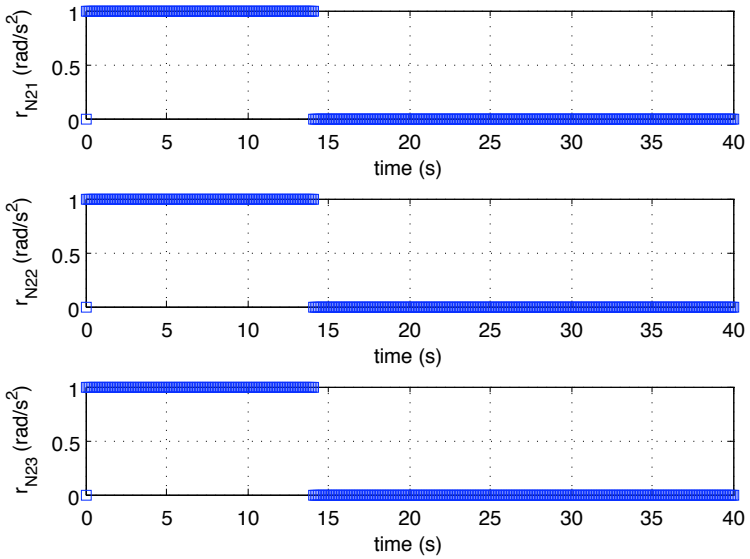

FIG. 7.9 – *Scenario 1 : residuals of mode* q_{N_2}

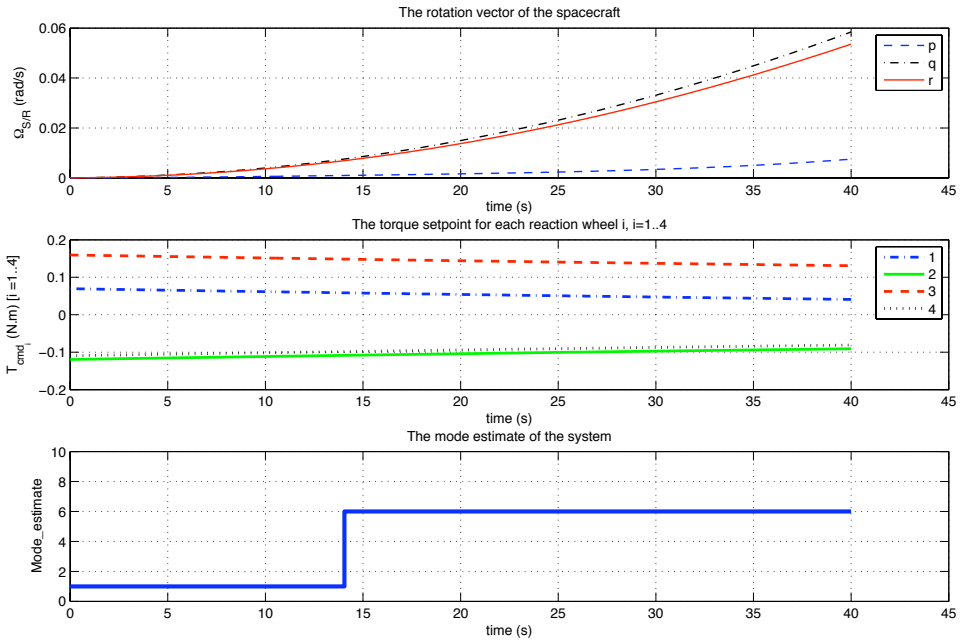

FIG. 7.10 – *Scenario 1 : mode estimation from observable variables*

ration modes is required to perform the wheel desaturation by activating thrusters for example (or other types of actuators).

7.5.2 Scenario 2

Now faults are injected to test the diagnosis module. Table 7.4 provides the discrete events issued from the abstraction of continuous dynamics required for scenario 2 (and 3). Scenario 2 considers the case in which the

Source mode	Destination mode	Associated event
q_{N_1} (1)	$q_{N_1F_1}$ (2)	$Ro_{N_1F_1}$
q_{N_1} (1)	$q_{N_1F_2}$ (3)	$Ro_{N_1F_2}$
q_{N_1} (1)	$q_{N_1F_3}$ (4)	$Ro_{N_1F_3}$
q_{N_1} (1)	$q_{N_1F_4}$ (5)	$Ro_{N_1F_4}$
q_{N_2} (6)	$q_{N_2F_1}$ (7)	$Ro_{N_2F_1}$
$q_{N_1F_1}$ (2)	$q_{N_2F_1}$ (7)	$Ro_{N_1N_2F_1}$

TAB. 7.4 – *Observable events issued from the abstraction of continuous dynamics*

system suffers the fault F_1, then transitions autonomously from $q_{N_1F_1}$ to $q_{N_2F_1}$.

$$q_{N_1} \xrightarrow{f_1 \text{ at } t=5s} q_{N_1F_1} \xrightarrow[\text{simulation time = } 40s]{T_{rot_3}<0.15\ N.m \text{ at t = } 15.7s} q_{N_2F_1}$$

Figure 7.11 shows the evolution of non observable variables : the motor torque T_{rot_i}, the kinetic momentum H_{rw_i} and the rotation velocity θ_i of each reaction wheel.

Figures 7.12, 7.13 and 7.14 show the evolution of filtered residuals of

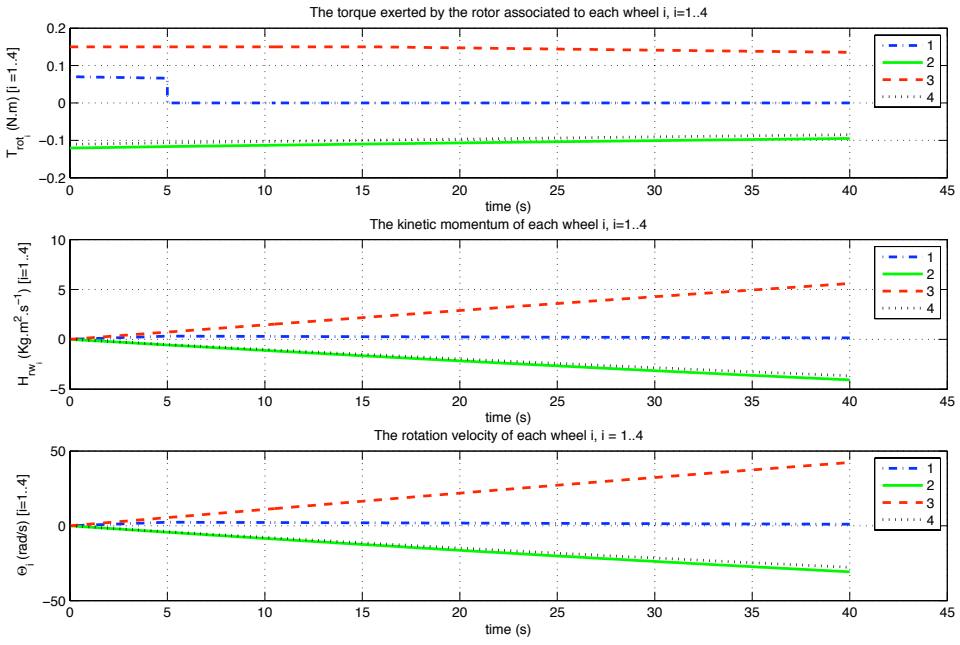

FIG. 7.11 – *Scenario 2 : non observables variables*

modes q_{N_1}, $q_{N_1F_1}$ and $q_{N_2F_1}$ during the simulation time.

At time $t = 5,01s$ residuals $[r_{N11}, r_{N12}, r_{N13}]$ of mode q_{N_1} change

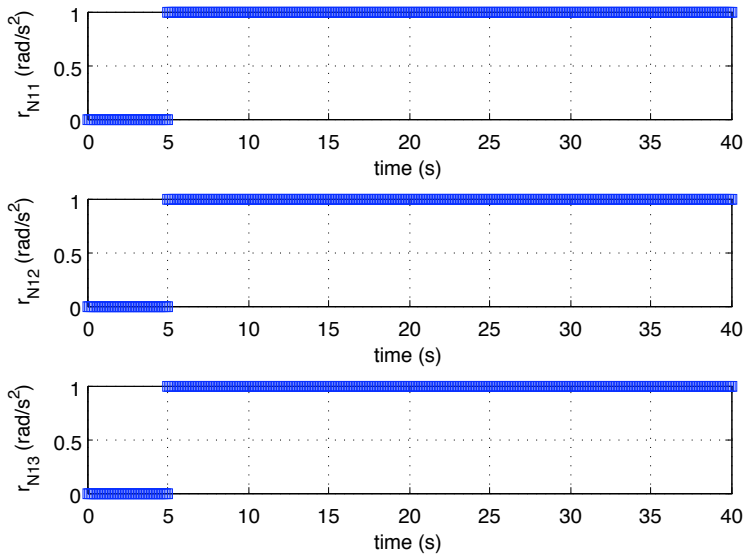

FIG. 7.12 – *Scenario 2 : residuals of mode* q_{N_1}

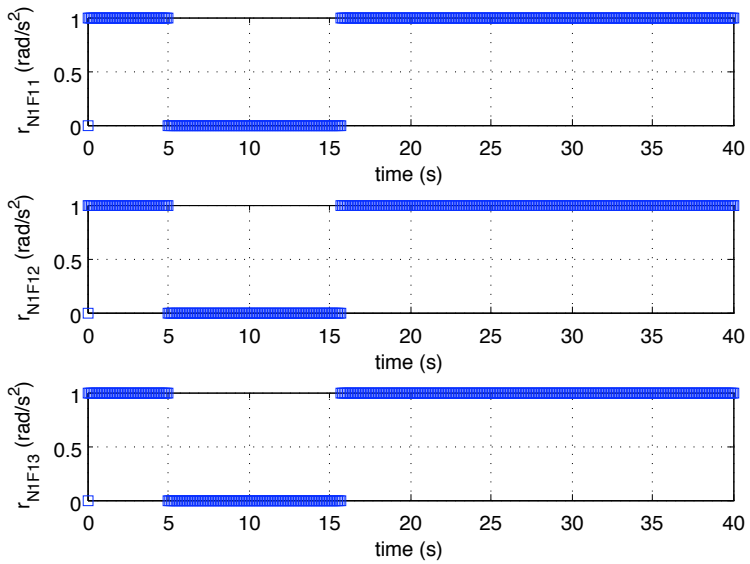

FIG. 7.13 – *Scenario 2 : residuals of mode* $q_{N_1F_1}$

their values from 0 to 1, and residuals $[r_{N1F11}, r_{N1F12}, r_{N1F13}]$ of mode $q_{N_1F_1}$ change their values from 1 to 0. Then, at $t = 15.7s$ residuals $[r_{N1F11}, r_{N1F12}, r_{N1F13}]$ of mode $q_{N_1F_1}$ change their values from 0 to 1 and residuals $[r_{N2F11}, r_{N2F12}, r_{N2F13}]$ of modes $q_{N_2F_1}$ change their values from 1 to 0. The time of the autonomous transition is detected by the diagnosis module.

The mode estimate and the observable variables are provided in Figure 7.15. Notice the mapping between the faulty mode transition shown in Figure 7.15 and the variation of rotation velocity (kinetic momentum) of

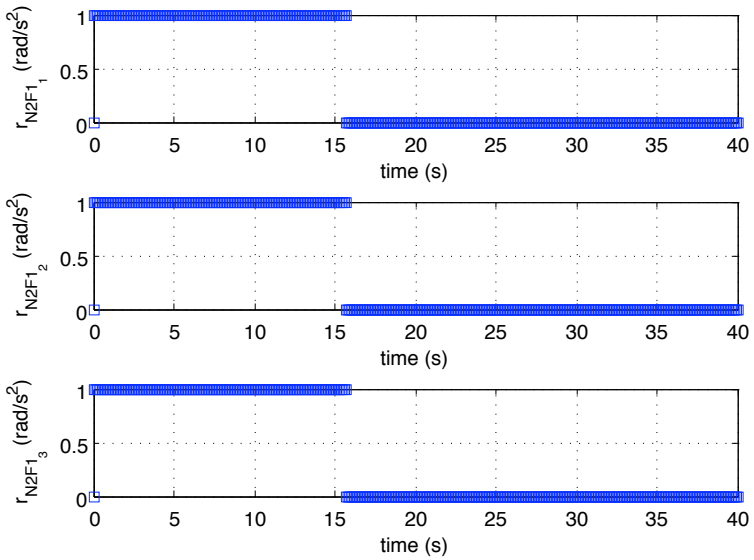

FIG. 7.14 – *Scenario 2 : residuals of mode* $q_{N_2F_1}$

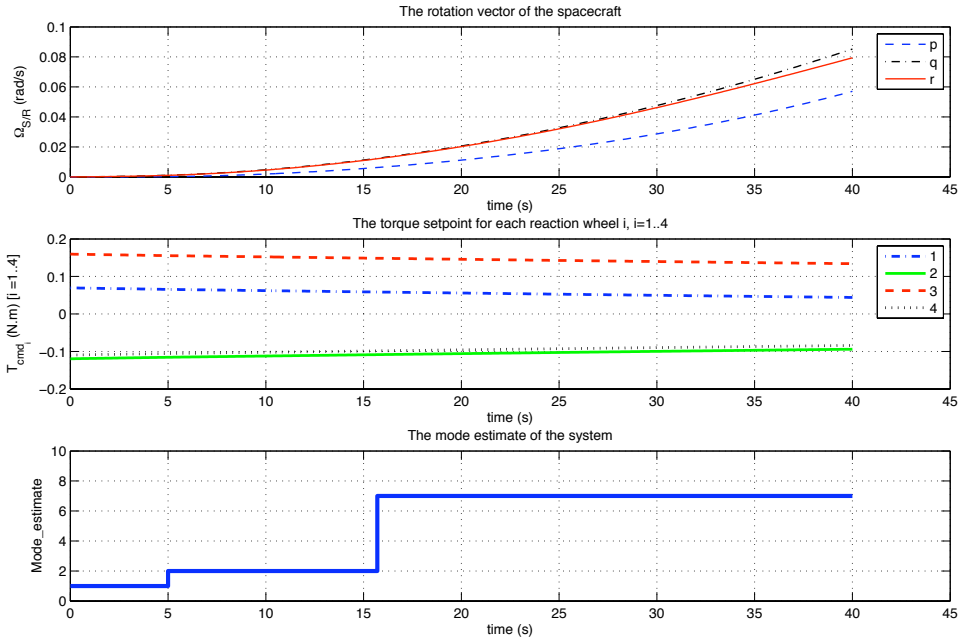

FIG. 7.15 – *Scenario 2 : mode estimation from observable variables*

wheel 1 : $\dot{\theta}_1$ (H_{rw_1}) shown in Figure 7.11. In the fault mode q_{N_2F1}, the failure of motor 1 translates into a deceleration of the wheel 1 down to 0 and an immediate cancellation of the motor torque T_{rot_1} as shown by Figure 7.11. The transition from the fault mode q_{N1F1} to the fault mode q_{N2F1} shown in Figure 7.15 is confirmed by Figure 7.11, in which the torque T_{rot_3} leaves the saturation region to the linear region.

7.5.3 Scenario 3

Scenario 3 considers the case in which the fault $F1$ occurs after the autonomous mode change.

$$q_{N_1} \xrightarrow{T_{rot_3}<0.15\ N.m \text{ at } t=14.05\text{ s}} q_{N_2} \xrightarrow{f_1 \text{ at } t=25s} q_{N_2F_1}$$
$$\text{simulation time} = 40s$$

The system switches autonomously from q_{N_1} to q_{N_2} at time $t = 14.05$ s, then suffers the fault F_1 at time $t = 25$ s. The evolution of non observable variables is provided in Figure 7.16. The residuals of concerned modes are

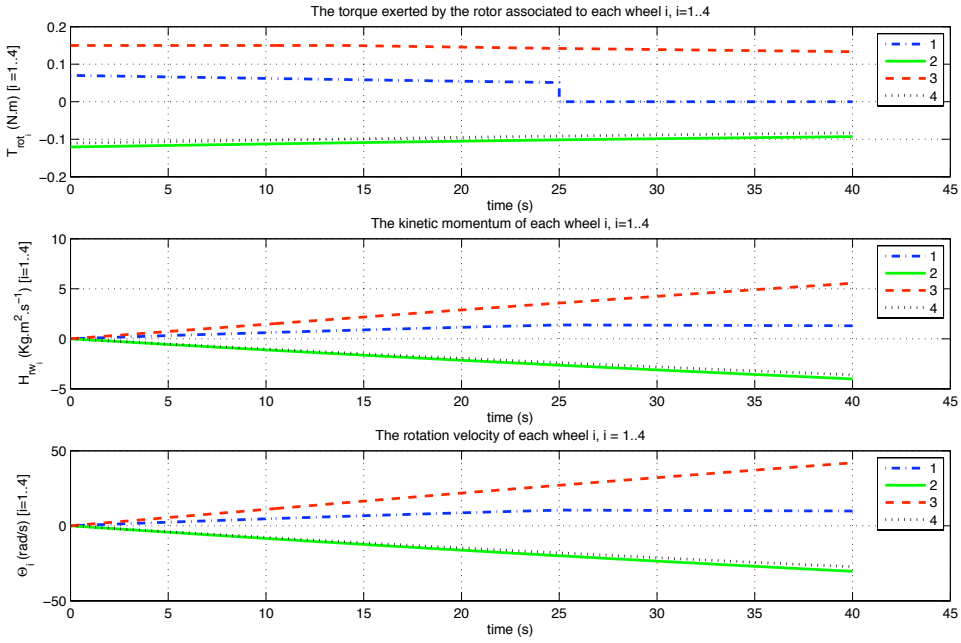

FIG. 7.16 – *Scenario 3 : non observables variables*

computed from observable variables and provided in Figures 7.17, 7.18 and 7.19. The mode estimate and the observable variables are provided in Figure 7.20. Let us notice the mapping between the faulty mode transition shown in Figure 7.20 and the variation of rotation velocity (kinetic momentum) of wheel 1 : $\dot{\theta}_1$ (H_{rw_1}) shown in Figure 7.16.

In conclusion, we notice that the occurrence of the fault $F1$ is detected in the nominal mode $N1$ (scenario 2) as well as in the nominal mode $N2$ (scenario 3). The diagnosis module tracks successfully transitions between nominal modes as well as transitions from nominal to fault modes.

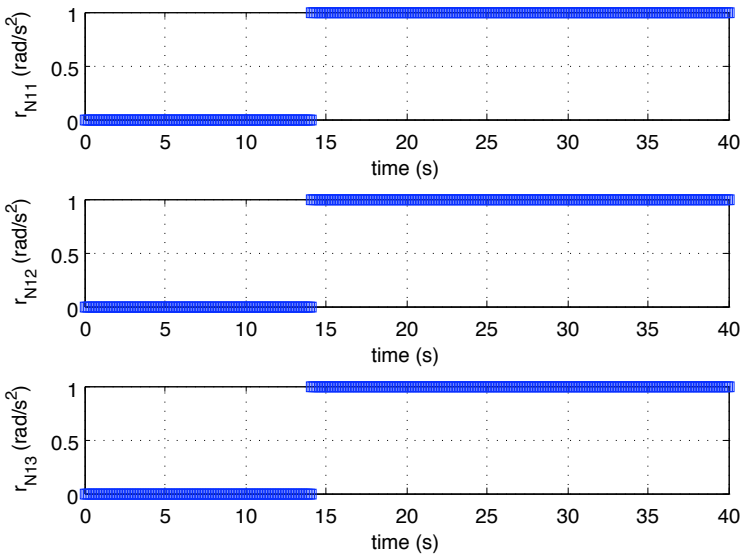

FIG. 7.17 – *Scenario 3 : residuals of mode* q_{N_1}

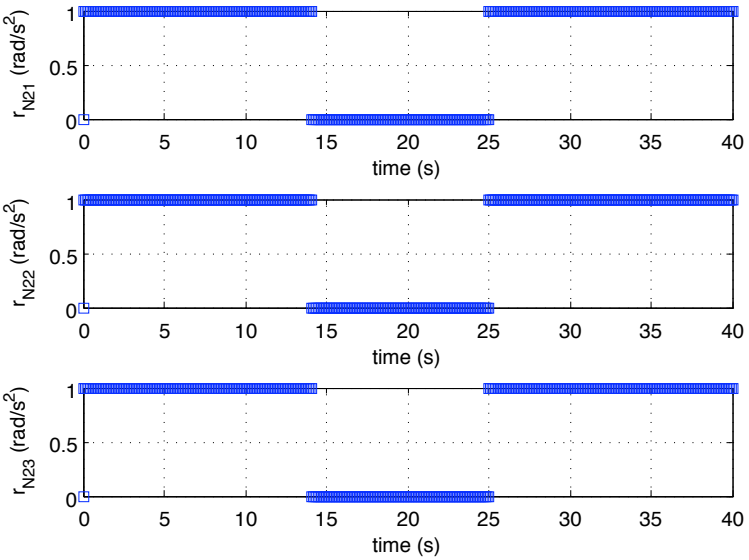

FIG. 7.18 – *Scenario 3 : residuals of mode* q_{N_2}

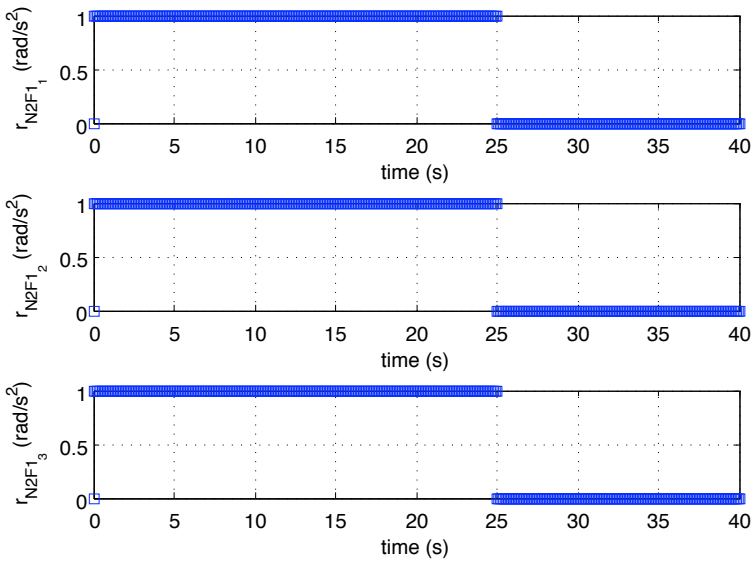

FIG. 7.19 – *Scenario 3 : residuals of mode* $q_{N_2F_1}$

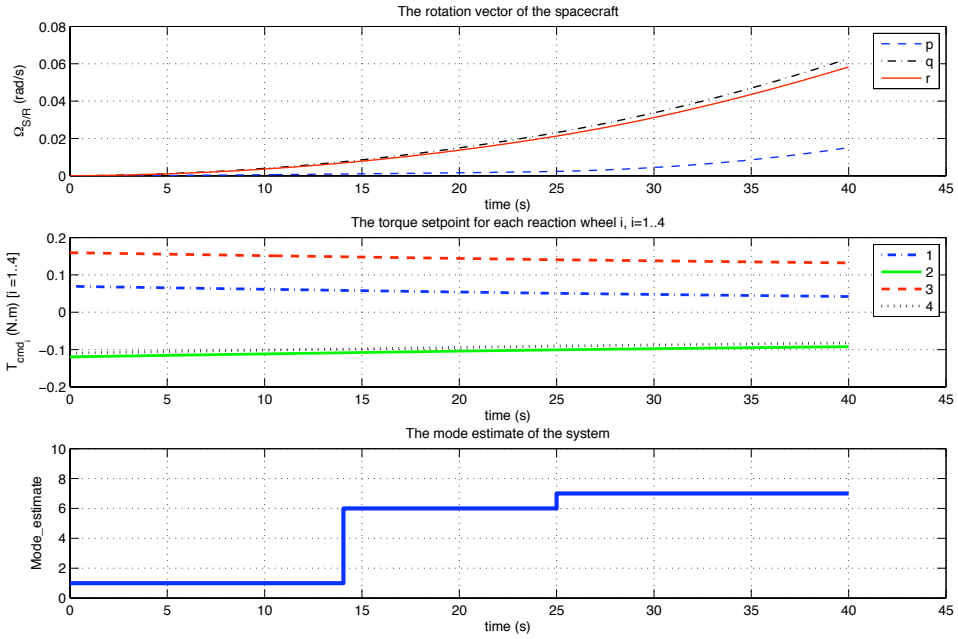

FIG. 7.20 – *Scenario 3 : mode estimation from observable variables*

Conclusion and Perspectives

This thesis addresses several aspects related to diagnosis in the framework of hybrid systems. The pursued goal is to provide a diagnosis method enhanced with an active diagnosis strategy that permits to disambiguate the output of the (passive) diagnosis module when it is not reduced to one only diagnosis hypothesis. The active diagnosis method relies on the diagnosability properties of the system. This is why a significant part of the work has been devoted to the characterization of diagnosability in the chosen hybrid modeling framework. This issue is quite novel and has been investigated by very few works. The same is true for the active diagnosis approach. We hence believe that the proposed methods are a valuable step forwards the analysis of diagnosability and diagnosis of hybrid systems and that they constitute a significant set of new results. The main contributions can be listed as follows :

- a hybrid modeling framework in which the behavior of the hybrid system is seen as the contribution of two underlying systems, a continuous multimode system – for which the transition function between modes is not constrained – and a discrete-event system. The behavioral representation is expressed by a hybrid automaton in which the discrete states correspond to the different behavioral modes of the system and every mode has a set of associated algebraic differential equations that represent its continuous behavior. The modeling framework supporting the work is quite appropriate for large systems. Considering operational modes allows one to decide how to organize the behavioral models and their dynamics. It offers the flexibility to represent non linear dynamics or to use piecewise linear models.
- a passive diagnosis approach that combines methods rooted in the artificial intelligence and the control fields. On the multimode system side, it is based on the definition of *mode signatures*, which rely on defining a set of residuals for each mode of the system. For this purpose, the parity space approach is extended to multimode systems. But the method does not presume of a specific method to generate the residuals and any other method could be used, as illustrated in our case study (cf. Chapter 7). The key idea of our hybrid diagnosis approach is then to abstract the mode signature changes in terms of a set of observable discrete events. These events are then considered together with the pure discrete events to define the alphabet of the language of the hybrid system. We define this language as well as the corresponding finite state generator called the *behavior automaton*. The diagnoser approach is then applied to the behavior automaton to perform on-line diagnosis.
- a definition for the diagnosability the underlying continuous mul-

timode system and a method for its analysis. The new concepts of *mirror signature*, *reflexive signature*, and *mode signature* are defined. The diagnosability of the multimode system can be achieved by the defined *mutual diagnosability* or 3^{rd} *diagnosability*. The necessary and sufficient criterion for multimode systems diagnosability is stated and proved, extending the work by Cocquempot *et al.* (2004).

- the characterization of the diagnosability of the hybrid system based on the language of the hybrid system. Two sufficient conditions for hybrid diagnosability are provided in terms of the the underlying multimode system on one hand and the underlying discrete system on the other hand. Finally, the necessary and sufficient criterion is given.
- an active diagnosis method, which is guided by the diagnosability properties of the hybrid system. This method considerable enhances the passive diagnosis approach, by allowing to disambiguate the diagnosis results when necessary. A new finite state machine called the *active diagnoser* is defined based on the new concepts of *controllable events*, *induced controllable events* and *controllable paths*. It is embedded in the diagnoser and defines the sets of states whose uncertainty can be reduced. The active diagnosis problem is formalized as a conditional planning problem and consists on exciting the hybrid system by applying active diagnosis control, observing the dynamic response (continuous or event-based) of the hybrid system and then deciding about the next sequence of actions. A mapping between the active diagnoser and an AND-OR graph is established. An active diagnosis plan is a controllable path in the AND-OR graph, starting from an ambiguous state and leading to a certain state. An algorithm is proposed to explore the AND-OR graph and returns active diagnosis plans. These plans can be guaranteed or not and lead to the new definitions of *active diagnosability* and *non guaranteed active diagnosability*. This new notions are compared to I-diagnosability introduced by Sampath *et al.* (1995). Diagnosability of the hybrid system w.r.t active diagnosis is also discussed.
- a demonstration of our approach on the attitude control system of a satellite, whose actuators are assumed to be composed of four reaction wheels. Thales Alenia Space provided a MATLAB/Simulink simulator for this case study. It was our job to derive all the mathematical equations from spatial mechanics and to organize them in a comprehensive framework to form a hybrid model. The continuous models associated to the different modes are generally non linear and the models were manipulated in order to generate appropriate residuals using an original and non standard method. Three application scenarios have been tested, providing good results and showing the relevance of our approach.

The proposed method remains very generic in the sense that it provides a general framework in which the procedures can be instantiated by different methods. For example, the residual generation procedure that has been illustrated with a parity space method in the thesis and with a specific method in the spatial attitude control system case study can be implemented by any other specific method for generating residuals. The same

is true for the event-based diagnoser method, that could be replaced by the twin plants method for instance Pencolé (2004). The active diagnosis approach is fully integrated in the diagnoser approach and significantly enhances the whole approach. It is quite generic as well, given that the formalization as a conditional planning problem permits to call for different planning methods.

The perspectives of this work can be foreseen in several directions. Some of them are in the short term and directly follow the achieved work. Others are openings that can be seen as wider undertakings.

Section 4.4 in Chapter 4 is an illustration of how the parity-based mode estimator could be coupled with filter-based Hybrid mode Estimator (HME) enhancing the hybrid state estimation. This is certainly a direction to go, particularly for pursuing and completing the active diagnosis method. Indeed, at the moment, the plan is given in terms of discrete actions and transitions. However, some of the transitions may be supported by guards involving continuous variables. Triggering such transitions hence means being able to apply the appropriate continuous control input that would drive the system to satisfy the guard. Hence, the continuous state variable estimation is required and the coupling with the HME is justified.

The connection that has been drawn between active diagnosability and I-diagnosability as defined by Sampath *et al.* (1995) is another interesting link. It would be another immediate work to consolidate this link and extend diagnosability to account for actions. The proposed framework follows a centralized approach that may result computationally inadequate. However, let us notice that the modeling is based on a hybrid automaton for which the discrete states correspond to the operational modes of the system. This approach is hence far from blowing up like standard discrete-event approaches in which the states of the automaton represent low level states of the system, defined with respect to behavioral variable changes. Now, an improvement at the modeling level would certainly be to use concurrent or communicating hybrid automata, each one representing the behavior of one component of the system. In this case, it would be interesting to devise the decentralized versions of the proposed methods.

Diagnosis as well as diagnosability analysis are based on mode signatures at the continuous level but then abstract their dynamics in terms of a specific set of discrete events. These discrete events are then considered together with the pure discrete events in the behavior automaton of the hybrid system. The framework then falls back into a discrete-event framework for which discrete-event methods are used, namely the diagnoser. Another interesting direction would be to use a unified signature framework as defined by Pucel (2008). Yet an interesting idea would be to define a new concept of *hybrid signature*, that would allow us to avoid the event abstraction phase and to work directly with the original hybrid automaton.

In conclusion, the framework proposed for active diagnosis promises to be very valuable. It is an interesting piece of theoretical work and provides very important perspectives from the application point of view.

Appendix

Chapter A

Contents

A.1 Determining the parity space order

As mentioned before, the underlying continuous model associated to every operating mode q_i is represented in the state space as follows :

$$\begin{cases} X_i(n+1) &= A_i X_i(n) + B_i U(n) + E_{x_i}\epsilon(n) \\ Y(n) &= C_i X_i(n) + D_i U(n) + E_{y_i}\epsilon(n) \end{cases}$$

with :
- $X_i(n)$: the state vector at time step nT_s.
- $U(n)$: the input vector at time step nT_s.
- $Y(n)$: the output vector at time step nT_s.
- $\epsilon(n)$: the noise vector at the time step nT_s.

T_s is the sampling period ; $A_i, B_i, C_i, D_i, E_{x_i}$ and E_{y_i} are constant matrices of appropriate dimensions.
Let consider the matrix $O_i^{p_i}$ is defined by :

$$O_i^{p_i} = \begin{pmatrix} C_i \\ C_i A_i \\ ... \\ C_i A_i^{p_i} \end{pmatrix}$$

To compute analytic redundancy relations using the parity space approach we need to determine a matrix $\Omega_i^{p_i}$ orthogonal to $O_i^{p_i}$.
$\Omega_i^{p_i}.O_i^{p_i} = 0 \Leftrightarrow (O_i^{p_i})^T.(\Omega_i^{p_i})^T = 0$
Since $O_i^{p_i}$ contains $m \times (p+1)$ rows and n columns, we have :
$m \times (p_i+1) = dim(kernel((O_i^{p_i})^T)) + rank((O_i^{p_i})^T) \Leftrightarrow$
$m \times (p_i+1) = dim(kernel((O_i^{p_i})^T)) + rank((O_i^{p_i}))$
The existence of $\Omega_i^{p_i}$ is guaranteed when $dim(kernel((O_i^{p_i})^T)) \geq 1$. In the other hand, we have $rank((O_i^{p_i})) \leq n$. Thus, in the worst case when we have $rank((O_i^{p_i})) = n$, there exists an integer p_i such that $(p_i+1) \times m \geq n$. Consequently, there always exists an order p_i such that $\Omega_i^{p_i}$ is orthogonal to $O_i^{p_i}$. In practice, we take to smallest order p_i such that guarantees this.
We can distinguish two cases :

A.1.1 Static Redundancy

When the the number of outputs m is bigger than the rank of C (the number of independent rows of C). This case models the ideal case when we have enough sensors to directly observe the state of the system. Consequently, we only need inputs and outputs at sampling time $n.T_s$ to obtain analytic redundancy relations and the order of the parity space is equal to 0. We say that we have *static redundancy*.

A.1.2 Dynamic Redundancy

When the number of outputs m is less (or equal) than the rank of C. It models the most general case. In this case the order of the parity space is at least equal to 1. It means that a temporal windows of is needed to observe inputs/outputs at sampling times $n, n-1, ..., n-p$. We say that we have *dynamic redundancy*. In our MATLAB/SIMULINK program, we compute the smallest p such that analytic redundancy relations exist. Indeed, the length of the observation windows has a direct implications on

the mode estimation delay. Nevertheless, the use of a bigger value (a non optimal value) of the parity space order (equal to the system order) can be useful and allows us to have more analytic redundancy relations that can improve the mutual diagnosability between system modes.

A.2 HYDIAG Softaware : Class Diagram

HYDIAG is a software developed in the context of this thesis for hybrid systems diagnosis. It takes as parameters :

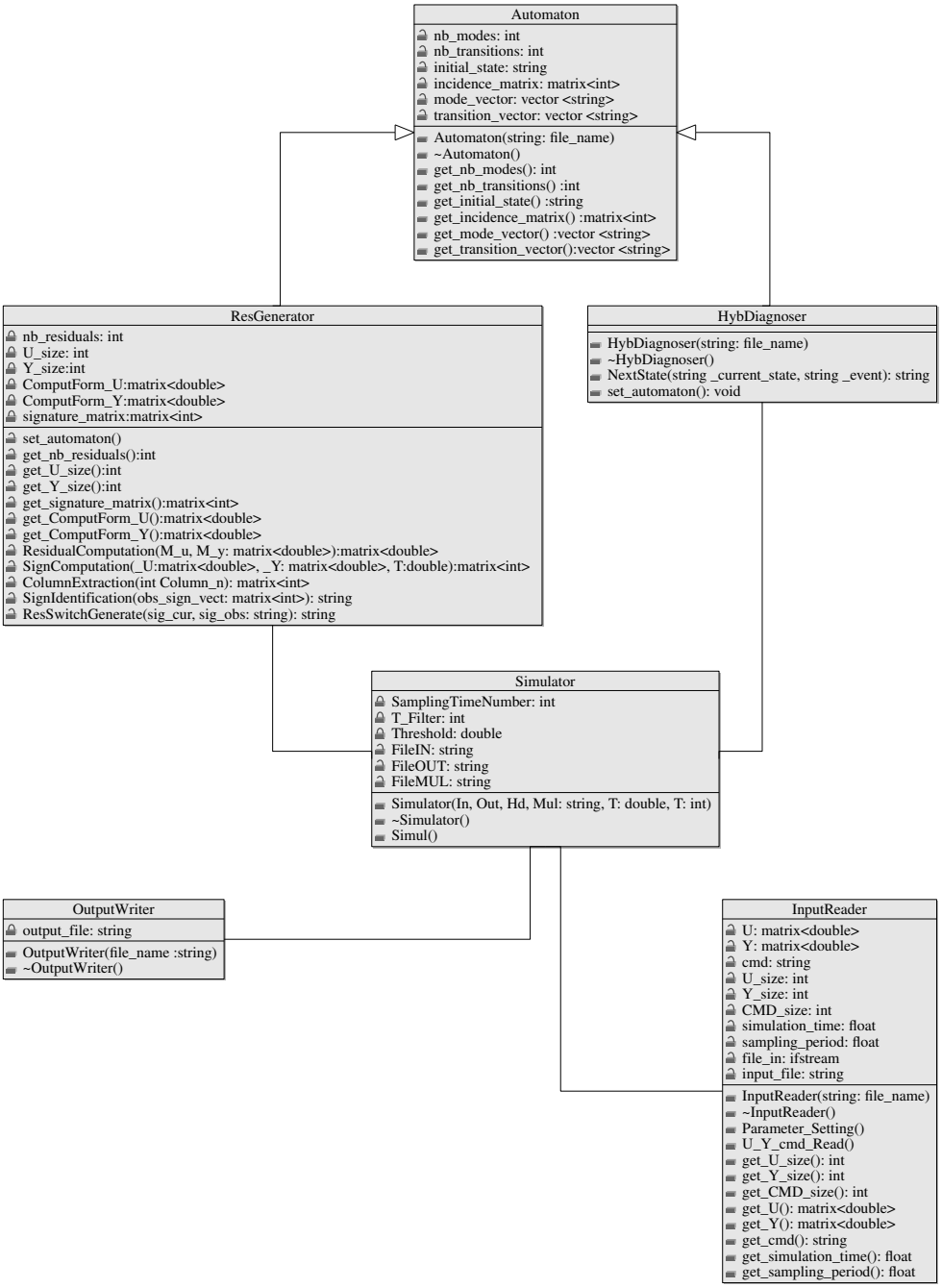

FIG. A.1 – *The UML Class diagram of HYDIAG software developed in C++*

- the structure of the hybrid diagnoser built from the behavior automaton (using DIADES from Pencolé (2006)) included in a file "*_NAME.hd*".
- the matrix modeling computation and evaluation forms of the ana-

lytic redundancy relations associated to operating modes included in a file "*_NAME.mul*".

The input file : "*_NAME.in*" contains the input/output and the observable discrete events at each sampling time. The simulation parameters (the sampling period, the simulation time, the size of continuous input/output vectors and observable discrete events) are specified in the beginning of this file. The simulator returns the mode estimate at each sampling time in the output file "*_NAME.out*". The software architecture is provided in Figure A.1.

Bibliography

M. Abramovici, C. Strond, C. Hamilton, S. Wijesuriya, and V. Verma. Using roving stars for on-line testing and diagnosis of fpgas infault-tolerant applications. In *Proceeding of the International Test Conference*, pages 973–982, Atlantic City, NJ (USA), 1999. (Cited page 75.)

O. Adrot, D. Maquin, and J. Ragot. Fault detection with model parameter structured uncertainties. In *Proceeding of the European Control Conference, ECC'99*, Karlsruhe, Germany, 1999. (Cited page 15.)

B. Bakhache and I.V. Nikiforov. Reliable detection of faults in measurement systems. *International Journal of Adaptive Control and Signal Processing*, 14(7) :683–700, 2000. (Cited page 10.)

M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice Hall Information and System Sciences Serie, 1993. (Cited pages 31 et 32.)

M. Basseville, M. Kinnaert, and M. Nyberg. On fault detectability and isolability. *European journal of control*, 7(6) :625–641, 2001. (Cited pages 31 et 32.)

M. Bayoudh and L. Travé-Massuyès. An algorithm for active diagnosis of hybrid systems casted in DES framework. *Accepted for presentation in the 2nd IFAC Workshop on Dependable Control of Discrete System*, 2009. (Cited page 82.)

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. Hybrid systems diagnosability by abstracting faulty continuous dynamics. In *Proceedings of the 17th International Workshop on Principles of Diagnosis DX'06*, pages 9–15, Burgos, Spain, 2006. (Cited page 41.)

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. State tracking in the hybrid space. In *Proceedings of the 18th International Workshop on Principles of Diagnosis DX'07*, pages 221–228, Nashville, TN, USA, 2007. (Cited page 55.)

M. Bayoudh, L. Travé-Massuyès, and X. Olive. Coupling continuous and discrete event system techniques for hybrid systems diagnosability analysis. In *Proceedings of the 18th European Conference on Artificial Intelligence ECAI*, pages 219–223, Patras (Greece), 2008. (Cited pages 33, 41 et 47.)

M. Bayoudh, L. Travé-Massuyès, and X. Olive. Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the 17th International Federation of Automatic Control, World Congress, IFAC-WC*, pages 7265–7270, Seoul (Korea), 2008. (Cited pages 39 et 41.)

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. Towards active diagnosis of hybrid systems. In *Proceedings of the 19th International Workshop on Principles of Diagnosis DX'08*, pages 231–237, Blue Mountains, Australia, 2008. (Cited pages 76 et 78.)

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. Active diagnosis of hybrid systems guided by diagnosability properties. *Accepted for presentation in the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009. (Cited page 78.)

E. Benazera, L. Travé-Massuyès, and P. Dague. State tracking of uncertain hybrid concurrent systems. In *Proceeding of the 13th International Workshop on Principles of Diagnosis DX'02*, pages 106–114, Semmering, Austria, 2002. (Cited page 22.)

E. Benazera. *Diagnosis and Reconfiguration based on Hybrid Concurrent Models - Application to Autonomous Satellites, PhD thesis of Paul Sabatier University*. Toulouse, France, 2003. (Cited pages 4 et 23.)

P. Bertoli, A. Cimatti, M. Roveri, and P. Traverso. Planning in nondeterministic domains under partial observability via symbolic model checking. *In Proceedings of the Seventeenth International Joint Conference on Artifical Intelligence*, pages 473–478, 2001. (Cited page 82.)

G. Biswas, M.O. Cordier, J. Lunze, L. Travé-Massuyès, and M. Staroswiecki. Diagnosis of complex systems : Bridging the methodologies of the FDI and DX communities. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special section*, 34(5) :2159–2244, 2004. (Cited pages 11 et 18.)

S. Biswas, D. Sarkar, S. Mukhopadhyay, and A. Patra. Diagnosability analysis of real time hybrid systems. *IEEE International Conference on Industrial Technology, (ICIT)*, pages 104–109, 2006. (Cited pages 32 et 33.)

L. Blackmore, S. Funiak, and B.C Williams. Combining stochastic and greedy search in hybrid estimation. *in Proceedings of the* 20^{th} *National Conference on Artificial Intelligence*, pages 282–287, 2005. (Cited page 57.)

H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33 :780–783, 1998. (Cited page 57.)

J. Chen and R.J. Patton. A re-examination of the relationship between. parity space and observer- based approaches in fault diagnosis. *In Proceedings of IFAC Safeprocess'04*, pages 590–596, 1994. (Cited page 31.)

E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7) :603–614, Jul 1984. (Cited pages 11 et 13.)

A. Cimatti, C. Pecheur, and R. Cavada. Formal verification of diagnosability via symbolic model checking. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'03*, pages 363–369, Acapulco, Mexico, 2003. (Cited page 27.)

V. Cocquempot, T. El Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. *IEEE/IFAC-ASCC : Asian Control Conference*, 2004. (Cited pages 33, 62 et 116.)

O. Contant, S. Lafortune, and D. Teneketzis. Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems*, 16(1) :9–37, 2006. (Cited page 27.)

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(52163-2177), 2004. (Cited pages 11 et 18.)

J de Kleer and B C Williams. Diagnosing multiple faults. *Artif. Intell.*, 32(1) :97–130, 1987. (Cited page 18.)

B. Dubuisson. *(sous la direction de) Diagnostic, Intelligence Artificielle et Reconnaissance des Formes*. Hermés, 2001. (Cited page 9.)

G.K. Fourlas, K.J Kyriakopoulos, and N.J. Krikelis. Diagnosability of hybrid systems. In *Proceedings of the 10th Mediterranean Conference on Control and Automation-MED2002*, Lisbon, Portugal, July, 2002. (Cited page 33.)

P.M. Frank. Fault diagnosis in dynamic systems using analytic and knowledge-based redundancy - a survey. *Automatica*, 26(3) :459–474, 1990. (Cited page 13.)

E. Frisk, D. Dustegor, M. Krys, and V. Cocquempot. Improving fault isolability properties by structural analysis of faulty behavior models : application to the damadics benchmark problem. In *Proceedings of the IFAC Safeprocess'03*, Washington, USA, 2003. (Cited page 31.)

J.J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Deker, 1998. (Cited page 31.)

R. Greiner, B.A. Smith, and R.W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis (research note). *Artificial Intelligence*, 41 :79–88, 1989. (Cited page 17.)

M. Grewal and K. Glover. Identifiability of linear and nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, 21(6) :833–837, Dec 1976. (Cited page 15.)

W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992. (Cited pages 15 et 75.)

T. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 278–292, New Brunswick, New Jersey, 1996. (Cited pages 21, 38 et 39.)

M. Hofbaur and B.C. Williams. Mode estimation of probabilistic hybrid systems. In *Proceedings of the 5th International Workshop on Hybrid Systems : Computation and Control, HSCC 02*, pages 253–266, London, UK, 2002. Springer-Verlag. (Cited pages 21, 22 et 24.)

M.W. Hofbaur and B.C. Williams. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B.*, 34(5) :2178–2191, 2004. (Cited pages 39, 56 et 57.)

J. Hopcroft, R. Motwani, and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Hardcover, 2000. (Cited page 27.)

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. (Cited page 27.)

P. Jimenez and C. Torras. An efficient algorithm for searching implicit and/or graphs with cycles. *Artificial Intelligence*, (124)1 :1–30, 2000. (Cited page 82.)

E. Kilic. Diagnosability of fuzzy discrete event systems. *Inf. Sci.*, 178(3) :858–870, 2008. (Cited page 27.)

D. Koller and U. Lerner. Sampling in factored dynamic systems. *Sequential Monte Carlo in Practice*, pages 445–464, 2001. (Cited page 22.)

X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich, and P. Cheung. Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems. In *Proceeding of the 40th IEEE Conference on Decision and Control*, pages 793–801, Orlando, Florida, USA, 2001. (Cited page 21.)

X. Koutsoukos, J. Kurien, and F. Zhao. Monitoring and diagnosis of hybrid systems using particle filtering methods. In *Proceedings of the Fifteenth International Symposium on the Mathematical Theory of Networks and Systems (MTNS'02), University of Notre Dame, South*, 2002. (Cited page 22.)

F. Liu and D. Qiu. Safe diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 53(5) :1291–1296, 2008. (Cited page 27.)

L.Lin and Y. Jiang. The computation of hitting sets : review and new algorithms. *Inf. Process. Lett.*, 86(4) :177–184, 2003. (Cited page 17.)

J. Lunze. Diagnosis of quantized systems based on a timed discrete-event model. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 30(3) :322–335, 2000. (Cited page 21.)

N. Lynch, R. Segala, F. W. Va, Nancy Lynch, Roberto Segala, Frits Vaandrager, H. B. Weinberg, and H. B. Weinberg. Hybrid i/o automata. *Lecture Notes in Computer Science, Hybrid Systems III*, 1066 :496–510, 1996. (Cited page 21.)

J.F. Magni and P. Mouyon. A generalized approach to observers for fault diagnosis. In *Proceeding of the 30th IEEE Conference on Decision and Control*, volume 3, pages 2236–2241, Brighton, United Kingdom, 1991. (Cited page 14.)

E. Manders, P. Mosterman, G. Biswas, and L. Barford. Transcend : A system for robust monitoring and diagnosis of complex engineering systems. pages 1–27, February, 1999. (Cited page 4.)

S.A. McIlraith, G. Biswas, D. Clancy, and V. Gupta. Hybrid systems diagnosis. In *Proceeding of the HSCC*, pages 282–295, Pittsburgh, PA, USA, 2000. (Cited page 22.)

E.A. Misawa and J.K. Hedrick. Non-linear observer - a state of the art survey. *Transactions of the ASME*, 111 :344–352, 1989. (Cited page 14.)

I. Morita and H. Okitsu. Signature frequency analysis for diagnosis of induction motor systems. *Electrical engineering in Japan*, 109(4) :102–112, 1990. (Cited page 10.)

S. Narasimhan and G. Biswas. An approach to model-based diagnosis of hybrid systems. In *Proceeding of the HSCC*, pages 308–322, Stanford, CA, USA, 2002. (Cited pages 21 et 22.)

U. Nayak and B.C. Williams. Fast context switching in real-time propositional reasoning. In *Proceedings of the AAAI-97*, pages 50–56, Palo Alto, California, USA, 1997. (Cited page 23.)

R.M. Neal, editor. *Bayesian learning for neural networks*. Number 118. Springer, Lecture Notes in Statistics, New York, 1996. (Cited page 10.)

M. Nicolaidis and Y. Zorian. On-line testing for vlsi-a compendium of approaches. *Journal of Electronic Testing*, 12(1-2) :7–20, 1998. (Cited page 75.)

H. Niemann. Fault tolerant control based on active fault diagnosis. In *Proceeding of the American Control Conference*, pages 2224–2229, Portland, OR, USA, 2005. (Cited page 76.)

H. Niemann. A setup for active fault diagnosis. *IEEE Transactions on Automatic Control*, 51(9) :1572–1578, 2006. (Cited page 76.)

M. Nyberg. Criterions for detectability and strong detectability of faults in linear systems. *International Journal of Control*, 75(7) :490–501, 2002. (Cited page 31.)

R. Patton and J. Chen. A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis. *European Journal of Diagnosis and Safety in Automation*, 1(2) :183–200, 1991. (Cited page 15.)

R. J. Patton, P. M. Frank, and R. N. Clarke, editors. *Fault diagnosis in dynamic systems : theory and application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989. (Cited page 15.)

Y. Pencolé. Diagnosability analysis of distributed discrete event systems. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 43–47, Valencia, Spain, 2004. (Cited pages 27 et 117.)

Y. Pencolé. Diades, diagnosis of discrete event systems, 2006. (Cited pages 53 et 123.)

A. Pouliezos, G. Stavrakakis, and C. Lefas. Fault detection using parameter estimation. *Quality and reliability engineering international*, 5 :283–290, 1985. (Cited page 15.)

X. Pucel. *A Unified Point of View on Diagnosability, PhD thesis of Toulouse University*. Toulouse, France, 2008. (Cited page 117.)

Z. Qiu and J. Gertler. Robust FDI and H_{inf} optimization. In *Proceeding of the 32nd IEEE Conference on Control and Decision*, San Antonio, Texas, USA, 1993. (Cited page 15.)

P. J. Ramadge and W. M. Wonham. The control of discrete-event systems. *Proc. IEEE*, 77(1) :81–98, 1989. (Cited pages 28, 49 et 76.)

R Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1) :57–95, 1987. (Cited pages 10, 15, 16 et 17.)

T. Rienmüller, M. Bayoudh, M. W. Hofbaur, and L. Travé-Massuyès. Hybrid estimation through synergic mode-set focusing. *Accepted for presentation in the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009. (Cited page 56.)

S. Russel and P. Norvig, editors. *Artificial Intelligence, A modern Approach, Second Edition*. Prentice Hall Series in Artificial Intelligence, 2003. (Cited pages 82 et 83.)

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. (Cited pages 27, 28, 29, 31, 33, 85, 86, 116 et 117.)

M. Sampath, S. Lafortune, , and D. Teneketzis. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control*, 43(7) :908–929, 1998. (Cited pages 76 et 77.)

R. Schalkoff, editor. *Pattern recognition. Statical, structural and neural approaches*. John Wiley Sons, 1992. (Cited pages 9 et 10.)

M. Staroswiecki and G. Comtet-Varga. Analytic redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37 :687–699, 2001. (Cited pages 11, 13 et 95.)

M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5) :687 – 699, 2001. (Cited page 13.)

M. Staroswiecki and P. Declerck. Analytic redundancy in non-linear interconnected systems by means of structural analysis. In *Proceeding of the IFAC/IMACS/IFORS Conference, AIPAC'89*, pages 23–27, Nantes, France, 1989. (Cited page 13.)

P. Struss. Testing for discrimination of diagnoses. In *Proceeding of the 5th International Workshop on Principles of Diagnosis DX'94*, pages 312–320, New Palttz (USA), 1994. (Cited page 75.)

W. Terrell, editor. *Some Fundamental Control Theory I : Controllability, Observability, and Duality*, volume 106. Mathematical Association of America, 1999. (Cited page 79.)

D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 50(4) :476–492, 2005. (Cited page 27.)

L. Travé-Massuyès and P. Dague. *Modèles et raisonnements qualitatifs*. Hermés, 2003. (Cited page 10.)

L. Travé-Massuyès, T. Escobet, S. Spanache, and X. Olive. Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 36(6) :1146–1160, 2006. (Cited page 31.)

V. Verma, S. Thrun, and R. Simmons. Variable resolution particle filter. *International Joint Conference of Artificial Intelligence*, pages 9–15, 2003. (Cited page 57.)

B. C. Williams and P. U. Nayak. A model-based approach to reactive self-configuring systems. In *Pproceedings of the AAAI*, pages 971–978, Portland, Oregon, USA, 1996. (Cited pages 4, 21 et 23.)

T. S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete- event systems. *IEEE Transactions on Automatic Control*, 47(9) :1491–1495, 2002. (Cited page 27.)

Notations

ACS	Attitude Control System
AFD	Active Fault Control
ARR	Analytic Redundancy Relation
CS	Continuous Systems
DES	Discrete Event Systems
FDI	Fault Detection and Isolation
FDIR	Fault Detection and Isolation and Reconfiguration/Recovery
HME	Hybrid Mode Estimator
MBD	Model Based Diagnosis

Active Diagnosis of Hybrid Systems Guided by Diagnosability Properties
Application to Autonomous Satellites

Abstract : Motivated by the requirements of the space domain in terms of on-board diagnosis and autonomy, this thesis addresses the problems of diagnosis, diagnosability and active diagnosis of hybrid systems. Supported by a hybrid modeling framework, a passive approach for model-based diagnosis mixing discrete-event and continuous techniques is proposed. The same hybrid model is used to define the diagnosability property for hybrid systems and diagnosability criteria are derived.
When the diagnosis provided by the passive diagnosis approach is ambiguous, active diagnosis is needed. This work provides a method for performing such active diagnosis. Starting with an ambiguous belief state, the method calls for diagnosability analysis results to determine a new system configuration in which fault candidates can be discriminated. Based on a new finite state machine called *the diagnoser*, the active diagnosis is formulated as a conditional planning problem and an AND-OR graph exploration algorithm is proposed to determine active diagnosis plans.
Finally, the diagnosis approach is tested on the Attitude Control System (ACS) of a satellite simulator provided by Thales Alenia Space. The diagnosis module is successfully tested on several fault scenarios and the obtained results are reported.
Keywords : Hybrid Systems, Model-Based Diagnosis, Fault detection and Isolation, Parity Space Approach, Diagnoser, Diagnosability, Active Diagnosis, Autonomous satellites, Attitude Control System.

Diagnostic Actif pour les Systèmes Hybrides Guidé par les Propriétés de Diagnosticabilité
Application aux Satellites Autonomes

Résumé : Motivée par les besoins du domaine spatial en termes de diagnostic embarqué et d'autonomie, cette thèse s'intéresse aux problèmes de diagnostic, de diagnosticabilité et de diagnostic actif des systèmes hybrides. Un formalisme hybride est proposé pour représenter les deux dynamiques, continues et discrètes, du système. En s'appuyant sur ce modèle, une approche de diagnostic passif est proposée en mariant les techniques des systèmes à événements discrets et des systèmes continus. Un cadre formel pour la diagnosticabilité des systèmes hybrides a également été établi proposant des définitions et des critères pour la diagnosticabilité hybride.
Suite à un diagnostic passif ambigu, le diagnostic actif est nécessaire afin de désambiguïser l'état du système. Cette thèse propose donc une approche de diagnostic actif, qui partant d'un état de croyance incertain, fait appel aux propriétés de diagnosticabilité du système pour déterminer la configuration où les fautes peuvent être discriminées. Une nouvelle machine à états finis appelée *diagnostiqueur actif* est introduite permettant de formaliser le diagnostic actif comme un problème de planification conditionnelle. Un algorithme d'exploration de graphes ET-OU est proposé pour calculer les plans de diagnostic actif. Finalement, l'approche de diagnostic a été testée sur le Système de Contrôle d'Attitude (SCA) d'un satellite de Thales Alenia Space. Le module de diagnostic a été intégré dans la boucle fermée de commande. Des scénarios de faute ont été testés donnant des résultats très satisfaisants.
Mots Clés : Systèmes Hybrides, Diagnostic à base de Modèles, Détection et Isolation de Fautes, Espace de Parité, diagnostiqueur, Diagnostic Actif, Satellites Autonomes, Système de Contrôle d'Attitude.