A decentralized FDI scheme for spacecraft: Bridging the gap between model based FDI research & practice

Saurabh Indra[1,2,3*], Louise Travé-Massuyès[1,2], and Elodie Chanthery[1,2]

[1]CNRS ; LAAS ; 7 avenue du colonel Roche, Toulouse, France
[2]Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ;Toulouse, France
[3]Centre National d'Études Spatiales, Toulouse

Abstract

This paper introduces a decentralized fault diagnosis and isolation architecture for spacecraft and applies it to the attitude determination and control system (ADCS) of a satellite. A system is decomposed into functional subsystems. The architecture is composed of local diagnosers for subsystems which work with local models. Fault ambiguities due to interactions between subsystems are resolved at a higher level by a supervisor, which combines the partial view of the local diagnosers and performs isolation on request. The architecture is hierarchically scalable. The structure of the ADCS is modelled as constraints and variables and used to demonstrate the decentralized architecture.

1. Introduction

Modern spacecrafts are complex systems, with extremely high requirements on reliability. In ground based systems, reliability can often be achieved through hardware redundancy. However designing aerospace systems involves trading off between tough competing requirements, with hardware redundancy very costly in terms of size, weight and complexity. Therefore only a few of the most critical components can usually be made physically redundant. Analytical redundancy can be a powerful alternative means of ensuring functional reliability. Analytical redundancy involves comparing the behaviour of a system with a model of its expected behaviour.
Fault detection and isolation (FDI) based on a model of the system, known as model based diagnosis (MBD) is one approach to using analytical redundancy to increase the reliability of a system. Reconfiguration actions can be initiated after the FDI phase. There is a wide gap between the theory of MBD and its adoption for real space missions due to lack of mission pull [1]. The costs associated with MBD stem largely from the high complexity of the algorithms, and the modeling effort involved in diagnoser design. Efforts towards bridging the gap between theory and practice should focus firstly on considering realistic fault scenarios in the design phase. Secondly, it might be worthwhile to work towards realistic goals in the short term and use the experience gained to guide the development of more ambitious MBD applications. In this way the cost-value tradeoff for adopting MBD for space vehicles might be made more favorable. Automatic monitoring of housekeeping data and constructing decision support systems for operators and astronauts are some such applications. These systems should be integrated into existing operational procedures. Thirdly, the research community should demonstrate their methods and algorithms on testbeds representative of real world scenarios. The testbeds developed in industry often are not openly available to the academic community. Investing time and effort in building up and sharing realistic testbeds and simulations can reap rich dividends over the long term.
Most complex systems can be decomposed functionally into subsystems. In the aerospace industry as in many others, the system integrator is responsible for defining the systems and the interfaces of subsystems which are then constructed and provided by sub suppliers. The diagnosis modules associated with the subsystems would also usually be designed by the sub suppliers.
This paper proposes a scheme for the decentralized diagnosis of space systems. The architecture is composed of local diagnosers working with local models of their subsystems, with their knowledge of the environment around them limited to information about which variables interface with other subsystems. The local diagnosers attempt to explain anomalies detected in their subsystems. The quantities exchanged with other subsystems are ignored and this might

 *Corresponding author, email: sindra@laas.fr

lead to ambiguities. These ambiguities are resolved at a higher level by a supervisory diagnoser. The architecture is hierarchically scalable, which means that local diagnosers of a level can act as supervisors for lower level diagnosers working on constituent components. The diagnoser takes into account a model of the system and also the anticipated faults in the design phase. The proposed diagnosis architecture is shown in figure 1 as applied to the subsystems of a satellite. In this paper the FDI scheme is applied to the attitude determination and control system (ADCS) of a satellite, with the attitude determination (ADS) and attitude control (ACS) considered as subsystems with local diagnosers, and a supervisory diagnoser at the global ADCS level.

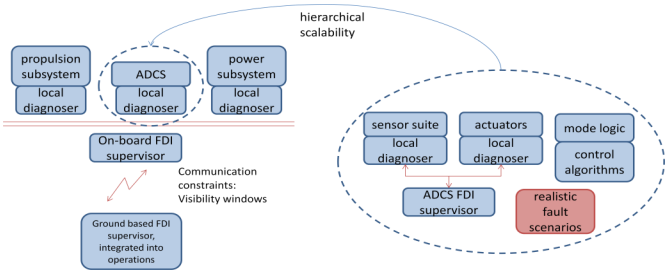

Figure 1: Decentralized diagnosis architecture applied to a satellite bus

The paper is structured as follows. Section 2 provides a summary of related work and positions the contribution of this paper. Section 3 discusses the background to the diagnosis method used and also the developed diagnosis architecture. The modelling of the ADCS both for simulation and for diagnoser design is presented in Section 4. As mentioned before, the diagnoser design method used takes into account the faults in the design phase. These are also summarized in section 3. The design of the decentralized diagnoser for the ADCS is described in section 5 along with diagnosis results. Section 6 concludes with a summary of the contribution and some perspective.

2. Related work

The model utilized by FDI algorithms can vary in framework and granularity. The present work deals with the decentralized diagnosis of systems modelled as continuous time systems. In particular we utilize the Analytical Redundancy Relation (ARR) approach to FDI within a structural framework. Such an approach is developed in [2] which describes an algorithm to analyse the structure of a system detecting redundant portions for use in ARR based diagnosis methods. This approach is further developed in [3] and [4]. While [3] includes information about interesting faults to increase the efficiency of the algorithm, [4] provides a transition from structural analysis to analytical computation of residual generators.

There has been considerable recent work aiming to apply model based FDI of continuous systems to aerospace systems and operations. Most of these works utilize kalman filter or observer banks to model the nominal and faulty behaviours of the system. The works discussed below make an attempt to include real world constraints and considerations into the design phase of the FDI module.

The design of a decision support system for automated monitoring of reaction wheel telemetry is illustrated in [5]. A kalman filter bank is used to detect and isolate faults with an Interacting Multiple Model (IMM) algorithm. A high fidelity reaction wheel model from [6] is utilized to demonstrate the effectiveness of the designed diagnoser. Realistic hard and soft faults are considered.

An observer based approach to the detection and isolation of gyroscope and horizon sensor faults is discussed in [7]. Matrix norms related to fault isolation are optimized to make the fault observer less sensitive to faults in actuators and momentum dumping torques.

An FDI module for the aerodynamic control surfaces of an atmospheric reentry vehicle is developed and demonstrated in [8]. An $H\infty/\ H_{-}$ robust approach is used to design residual generators. Faults for the flap actuators of the HL-20 reentry vehicle are diagnosed in the autolanding phase of the mission. A complete design, analysis and test cycle is demonstrated. The developed FDI approach can successfully handle realistic measurement noise and atmospheric disturbance profiles. Realistic fault scenarios for the actuators are considered. Performance indices critical for

technology adoption such as detection delay, complexity and computation requirements are used to demonstrate the viability of the proposed method.

[9] describes the development of a fault tolerant attitude determination system (ADS). Sensed data from rate sensors and vector sensors is fused with a linear Kalman filter. This attitude determination filter also estimates the rate gyro bias errors. The fault detection and diagnosis architecture utilizes Extended Kalman Filters (EKF) to deal with the possible nonlinearities introduced by faults. The FDI architecture is split into three stages corresponding to fault detection, preliminary isolation between rate and vector sensors and final diagnosis of faulty sensor. By designing the preliminary isolation phase, the computational requirement of the process is reduced leading to a faster diagnosis of the faulty sensor. Both single and multiple fault scenarios are handled. The developed ADS is able to provide reliable attitude estimates even if one of the rate sensors or vector sensors is faulty.

A fault detection method is developed in [10] for the Delfi N3Xt satellite currently being developed at the Delft University of Technology. An unscented Kalman filter is used to generate estimates for comparison with sensed values. While faults on the rate sensors could be detected, faults on the sensors linked to the quaternion calculation could not be precisely isolated.

A robust FDI approach for the thruster faults in the Mars Express orbiter is developed in [11]. A description of the MEX orbiter structure and the uncertainty and disturbances sources is provided. Also, the FDIR mechanism currently implemented on the spacecraft is introduced. The developed diagnoser is based on observer banks for FDI which are structurally decoupled from disturbances and estimated uncertainties. A detailed simulation of the orbiter including structural flexible modes, and realistic disturbances is utilized as a test bed. Performance indices to gauge the effectiveness of the diagnoser are introduced. These indices are used to compare the developed FDI scheme with that currently implemented on the MEX orbiter.

The contribution of the present work is the design of a decentralized diagnosis architecture. There has not been much work on the decentralized diagnosis of continuous systems. Algorithms for the diagnosis of continuous systems are adapted for our decentralized architecture. A similar approach for the diagnosis of systems modeled in a qualitative framework was introduced in [12]. All the work discussed above deals with FDI for either the ADS or ACS components along a centralized approach. In our work we consider the ADS and ACS both as subsystems of an ADCS and apply our decentralized diagnosis framework to resolve possible ambiguities between faults on the components which constitute the subsystems. The fault scenarios to be considered are taken into account in the design phase of the diagnoser. An architecture is developed which will be extended for the decentralized diagnosis of hybrid systems as part of future work.

3. A Decentralized Diagnosis Architecture

This section begins with a summary of the theoretical background to the diagnosis approach we use. We then proceed to introduce some notions required for extension of the ARR design approach to decentralized diagnosers. The diagnoser architecture is explained next, together with how the diagnoser is designed and implemented.

3.1 Background of diagnosis algorithms

Our approach to diagnosis is based on designing *residual generators* based on *structural redundancies* in the system. Residual generators are derived based on analytical redundancy relations which involve only observed quantities of the system. A residual generator takes as input the values of the observed variables and, in an ideal case, gives a non-zero output only in case the system behaviour is inconsistent with the model. Most of this development follows that in [2], [3] and [13].

Let the system description consist of a set of n equations involving a set of variables. The set of variables is partitioned into a set Z of n_Z known (or observed) variables and a set X of n_X unknown (or unobserved) variables. We refer to the vector of known variables as z and the vector of unknown variables as x.

We consider a *model*, denoted $M(z, x)$ or M for short, to be any set of equations relating the known variables z and the unknown variables x. The equations $m_i(z, x) \subseteq M(z, x)$, $i = 1, \ldots, n$, are assumed to be differential or algebraic

equations in z and x.

We say that a model $M(z, x)$ is consistent with a given trajectory of z, or concisely, consistent with z, if there is a trajectory of x such that the equations $M(z, x)$ are fulfilled.

Definition 1 (ARR for $M(z, x)$[14]) *Let $M(z, x)$ be a model, then an equation $r(z, \dot{z}, \ddot{z}, \ldots) = 0$ is an* ARR for $M(z, x)$ *if for each z consistent with $M(z, x)$, the equation is fulfilled.*

An ARR can be used to check if the observed variables z are consistent with the model and can be used as the basis of residual generators as defined below.

Definition 2 (Residual Generator for $M(z, x)$[14]) *A system taking a subset of the variables z as input, and generating a scalar signal r as output, is a* residual generator for the model $M(z, x)$*, if for all z consistent with $M(z, x)$, it holds that* $\lim_{t\to\infty} r(t) = 0$.

The *structure* of the system can be abstracted as a representation of which variables are involved in the different equations which make up the model of the system. This abstraction allows us to study the diagnosability properties independently of the linear or nonlinear nature of the systems. However it must be kept in mind that results obtained with such a structural representation are a best case scenario. Causality considerations and the presence of algebraic and differential loops determine which structural redundancies can be exploited for the design of residual generators.

Obtaining ARRs for a model $M(z, x)$ involves the elimination of unobserved variables, which can be infered from the bipartite graph. The bipartite graph indeed represents which unobserved variables are involved in the equations modeling the system. It can be shown [15] that ARRs correspond to so called complete matchings between X and M on the bipartite graph $G(M \cup X \cup Z, \mathcal{A})$, or equivalently on $G(M \cup X, A)$, where $A \subseteq \mathcal{A}$ and A is a set of arcs such that $a(i, j) \in A$ iff variable x_i is involved in relation m_j. A complete matching between X and M denoted by $\mathcal{M}(X, M)$, or $\mathcal{M}$ when there is no ambiguity, can be seen as a way to calculate the unobserved variables using the observed quantities. Equivalently, ARRs correspond to minimal structurally over determined (MSO) sets, which are sets of equations of the system with one more equation than unknowns [2]. Unobserved variables can be solved for using the set of equations, and then the one redundant equation can be used to check for consistency. We adopt an MSO set based ARR design method for our decentralized diagnoser architecture. However, for proving the equivalence of centralized and global diagnosers, we use the complete matching on a bipartite graph view on ARRs.

We can analyze the structural properties of a system modeled as a set of equations by using the canonical Dulmage-Mendelson (DM). This decomposition of a system model M results in the division of the model into three parts, the structurally overdetermined part represented by M^+, which has more equations than unknowns, the structurally just determined part represented by M^o and the structurally under determined part represented by M^-. The sets defined below formalize the notion of a structurally overdetermined set.

Definition 3 (Structurally Overdetermined equation set (SO) [2]) *A set M of equations is structurally overdetermined if M has more equations than unknowns.*

Definition 4 (Proper Structurally Overdetermined equation sets (PSO) [2]) *An SO set M is a proper structurally overdetermined (PSO) set if $M = M^+$.*

A PSO set is generically a testable subsystem, but it may contain smaller PSO subsets that are also testable subsystems. The minimal PSO sets, namely the MSO sets, are of special interest since they are at the core of the isolability properties.

Definition 5 (Minimal Structurally Overdetermined equation sets (MSO) [2]) *An SO set is a minimal structurally overdetermined (MSO) set if no proper subset is an SO set.*

An efficient algorithm to compute all possible MSO sets for a system is developed in [2]. However the number of possible MSO sets increases exponentially with the redundancy present in the system as measured by the difference between the number of equations and the number of unknown variables. The redundant equation sets which need to be exploited to construct residual generators can be limited to those which correspond to realistic or interesting faults. [3] introduces the concept of test equation supports (TES) which are sets of equations which express redundancy specific to a set of considered faults. Each TES corresponds to a set of faults which influence the residual generator constructed from the TES. This set of faults is known as the *test support* (TS). The corresponding quantities expressing minimal

redundancies are denoted minimal TES (MTES) and minimal TS (MTS). The set of MTES can be seen as a subset of the set of MSOs for the system corresponding to the set of interesting faults. An algorithm for finding MTES and MTS for a given system structural description and set of interesting faults is developed by modifying the MSO algorithm of [2]. We use $F(M)$ to denote the set of faults that influence any of the equations in M.

Definition 6 (Test Support(TS) [3]) *Given a model M and a set of faults F, a subset of faults $\zeta \subseteq F$ is a test support if there exists a PSO set $M' \subseteq M$ such that $F(M') = \zeta$.*

Definition 7 (Minimal Test Support (MTS)[3]) *Given a model, a test support is a minimal test support (MTS) if no proper subset is a test support.*

Definition 8 (Test Equation Support (TES)[3]) *An equation set M is a test equation support if M is a PSO set, $F(M) \neq \emptyset$, and for any $M' \supseteq M$ where M' is a PSO set it holds that $F(M') \supseteq F(M)$.*

Definition 9 (Minimal Testable Equation Support (MTES)[3]) *A TES M is a minimal TES if there exists no subset of M that is a TES.*

An MSO set or MTES signifies the theoretical presence of a structural redundancy which could be used to develop a consistency check for a part of the system. The corresponding MTS represents the faults which can be detected with this consistency check. In this was the MTS sets characterize the maximum possible fault isolability. Whether a residual generator can be analytically derived depends upon the causality restrictions on the equations in the set and the presence of algebraic and differential loops. We use in our work the residual generator derivation method proposed in [4]. This method relies on developing a computational sequence to successively solve for the unknown variables involved in an equation set. One redundant equation together with the developed computational sequence constitute a sequential residual generator. An algorithm to develop the computational sequence is provided.

The FDI scheme for a centralized case can be seen in figure 2.

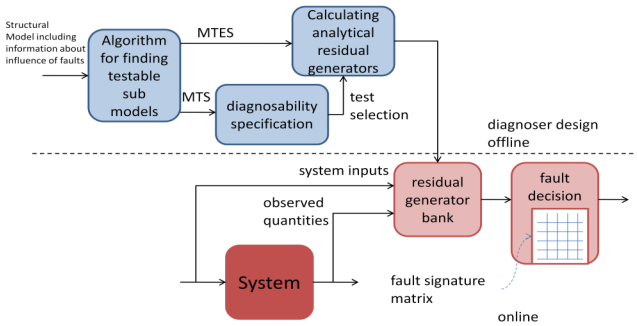

Figure 2: The design and implementation scheme for a centralized global diagnoser

After offline design, the diagnoser is implemented as a residual generator bank. The fault identification is carried out after fault detection using fault signatures which are vectors composed of the binary residual bank output (fault/no fault 1/0).

We now introduce the notions needed to decentralize the design and implementation of a diagnoser such as that of figure 2.

3.2 Notions for decentralized diagnosis

This section introduces the notions we need in order to devise the proposed decentralized architecture. First we introduce the decomposition of a global system into a set of sub-systems. Then we define a matching at the local level, the global level and at the supervisory level.

Hypothesis 1 *A decomposition of a system M, with associated bipartite graph $G(M \cup X \cup Z, A)$, into several subsystems M_i corresponds to a partition of its equations.*

Formally, let $M = \{M_1, M_2, \ldots, M_n\}$ with $M_i \subseteq M$

- $M_i \neq \emptyset$
- $\bigcup M_i = M$
- $M_i \cap M_j = \emptyset$ if $i \neq j$

Definition 10 (Variables of a subsystem i) *Considering $G(M \cup X \cup Z, A)$, we define X_i (Z_i) as the subset of vertices of X (Z) that are adjacent to some vertices in M_i, i.e*

$$X_i = \{u \in X : \exists v \in M_i, (u, v) \in A\}$$

$$Z_i = \{u \in Z : \exists v \in M_i, (u, v) \in A\}$$

The decomposition of the global system into several sub-systems leads to n subsystems denoted $M_i(x_i^{local}, z_i)$, with associated subgraphs $G(M_i \cup X_i^{local} \cup Z_i, A_i)$, $i = 1, \ldots, n$, where X_i^{local} is defined below.

Definition 11 (Local variables) *We define X_i^{local} as the subset of vertices of X_i that are adjacent only to some vertices in M_i, and not to some vertices of M_j, $j \neq i$, i.e*

$$X_i^{local} = \{u \in X_i : \nexists j (j \neq i) v \in M_j, (u, v) \in A\}$$

Lemma 1 $X_i^{local} = X_i \setminus (\bigcup_{j=1, j\neq i}^{n} (X_i \cap X_j))$

Definition 12 (Shared variables) *We define X^{shared} as the subset of vertices of X that can not be considered as local variables for any sub-system i.e*

$$X^{shared} = X \setminus (\bigcup_{i=1}^{n} X_i^{local})$$

Lemma 2 *By definition, $\forall i (1, \ldots, n), X_i^{local} \cap X^{shared} = \emptyset$.*

Definition 13 (Local complete matching) *A local complete matching $\mathcal{M}_i$ is a complete matching between X_i^{local} and M_i on the graph $G(M_i \cup X_i^{local}, A_i)$.*

Definition 14 (Global complete matching) *A global complete matching $\mathcal{M}$ is a a complete matching between X and M on the graph $G(M \cup X, A)$.*

Definition 15 (Hierarchical relation) *Let us consider the local subsystem graphs $G(M_i \cup X_i^{local}, A_i)$, $i = 1, \ldots, n$, and assume a local complete matching $\mathcal{M}_i$ exists for each of them. Also consider the set of relations that are not matched in any local complete matching $\mathcal{M}_i$. Let r be one of these relations. By construction, r relates a set of variables, whose unknown variables belong to only one of the X_i^{local} and possibly to X^{shared}. With $\mathcal{M}_i$, it is possible to substitute every variable included in X_i^{local} in r [1], so as to get a new relation r′ involving only unknown variables in X^{shared}. The new relation r′ is to be transferred to the upper level and is called a* hierarchical relation. *r is called the* source relation *of r′. The set of such relations is denoted R′.*

Definition 16 (Hierarchical complete matching) *A hierarchical complete matching $\mathcal{M}_h$ is a complete matching between X^{shared} and R′ on the graph $G_h(R' \cup X^{shared}, A')$.*

3.3 The equivalence of centralized and decentralized diagnosis

When designing decentralized diagnosers for a system, it is interesting to investigate any change in the diagnosability properties due to the decentralization. In particular we would wish that properties such as detectability and isolability of faults are not altered by decentralization. This can be ensured if the set of ARRs derived in the global and decentralized scenarios are identical. This section formalizes this equivalence, and provides the basis of the proof. A detailed discussion and proof can be found in [19].

Proposition 1 *Let M be a system and $\{M_1, M_2, \ldots, M_n\}$ be a decomposition of M, then the set of ARRs that can be derived (in a centralized way) for M is identical to the set of ARRs that can be derived in a decentralized way, i.e. deriving the ARRs for every subsystem M_i and for the hierarchical system composed of the hierarchical relations.*

This proposition is proved by showing that there exists a global complete matching if and only if there exist local complete matchings and a hierarchical complete matching and that these matchings lead to identical ARRs.

[1] *substitute* refers to replacing the variable along the calculation chain defined by the complete matching up to known variables.

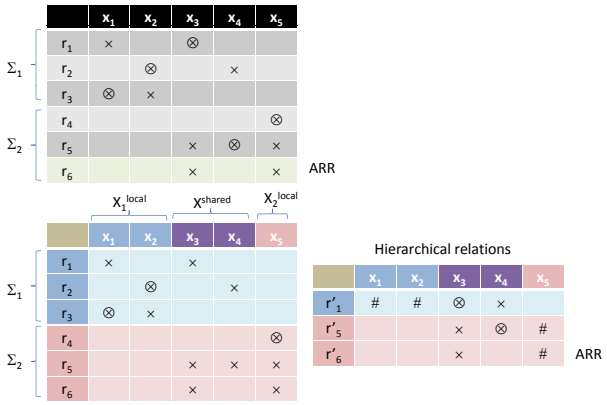

Figure 3: From a global system to a distributed system

3.3.1 "If" proof: from global to local

Proposition 2 *Let $\mathcal{M}$ be a global complete matching on $G(M \cup X, A)$ that leads to a set of ARRs that is non void, then for any decomposition into sub-systems $\{M_1, M_2, \ldots, M_n\}$, it is possible to find a set of local complete matchings $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n\}$ and a hierarchical complete matching $\mathcal{M}_h$ that leads to the same non-void set of ARRs.*

Proof idea: Suppose that $\mathcal{M}$ is a global complete matching of the system. When we decompose the system into subsystems, each relation that is matched with a shared variable in $\mathcal{M}$ is now available for being a hierarchical relation. This means that at the hierarchical level, each shared variable can be matched to the hierarchical relation whose source relation is the one it was matched to in $\mathcal{M}$. Consequently the matchings $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n\}$ lead to the same ARRs.

Figure 3 shows the decomposition of a system into 2 subsystems, and the resulting matchings. The global system represented by the adjacency matrix of $G(M \cup X, A)$ has 6 relations $r_1, \ldots, r_6$ and 5 variables $x_1, \ldots, x_5$. The system is decomposed into two subsystems Σ_1 and Σ_2, with $R_1 = \{r_1, r_2, r_3\}$ and $R_2 = \{r_4, r_5, r_6\}$. We can thus define $X_1^{local} = \{x_1, x_2\}$, $X_2^{local} = \{x_5\}$ and $X^{shared} = \{x_3, x_4\}$. At the top of Figure 3, there is the global complete matching marked by the relations with circles. At the bottom, we show the local complete matchings, for subsystems Σ_1 and Σ_2 on the left table and the resulting hierarchical relations r'_1, r'_5 and r'_6 on the right side. The # indicate the substituted variables in the hierarchical relations. The hierarchical complete matching is marked by the circles. One can notice that shared variables x_3 and x_4 are matched to r'_1 and r'_5, respectively, by $\mathcal{M}_h$ as they were to the source relation r_1 and r_5 by $\mathcal{M}$.

3.3.2 "Only if" proof: from local to global

Proposition 3 *Let $\{M_1, M_2, \ldots, M_n\}$ be the decomposition of a system into a set of n subsystems. Suppose that we have $(\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n)$ the set of local complete matchings for each subsystem represented by $G(M_i \cup X_i^{local}, A_i)$, and $\mathcal{M}_h$ the hierarchical complete matching on $G_h(R' \cup X^{shared}, A')$, then it is possible to find a global complete matching $\mathcal{M}$ on $G(M \cup X, A)$ that leads to the same set of ARRs.*

Proof idea: A hierarchical complete matching implies the existence of either a complete matching at the global level i.e. on $G(M \cup X, A)$, or of a set of substitution paths in either of subsystems which allows the matching of the shared variables by substitution. The set of relations involved in the local and hierarchical matchings can be shown to be exactly the same as that involved in the global complete matching.

3.4 Diagnoser architecture

Our decentralized diagnosis architecture is composed of a supervisory diagnoser for a system with local diagnosers for the subsystems composing the systems. We aim to keep the structure hierarchically scalable as shown in figure 1. There is no communication between diagnosers at a level. Diagnosers communicate between levels, with their supervisory diagnoser and the local diagnosers below them in the hierarchy. We aim to expose as little information as possible

about the subsystems. This is in keeping with our aim of achieving a decentralized architecture and also fits well into an integrator-subsystem supplier relationship.

The diagnosis process is explained below. The diagnoser design and implementation steps of the diagnosis process are explained with the help of figure 4 which can be considered as the decentralized counterpart of figure 2. We consider the diagnoser for a subsystem at level i of the diagnoser hierarchy. The communication required between diagnoser levels is highlighted in the explanation.

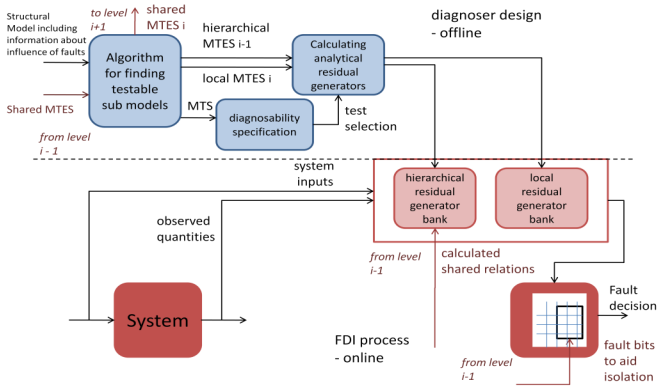

Figure 4: The design and implementation scheme of a decentralized diagnoser for a subsystem at level i

3.4.1 Decentralized diagnoser design

The diagnoser design is done *offline* and consists of the steps below. These steps are performed for each subsystem $M_{i,j}$ $j = 1 \ldots n_i$ at each level $i = 1 \ldots n_l$, with a nested loop. Here i signifies the level in the hierarchy, and j the enumeration of subsystem at that level.

1. *Use the MTES algorithm with the structural model of the subsystem $M_{i,j}$ as input*
   Output:
   (a) *local MTES for the subsystem $M_{i,j}$*
   (b) *MTS for the subsystem $M_{i,j}$*
   (c) *shared MTES for the subsystem $M_{i,j}$*
2. *Store shared MTES for supervisory diagnoser design at level $i + 1$*
3. *Use the MTES algorithm with the shared MTES of subordinate local diagnosers at level $i - 1$*
   Output:
   (a) *hierarchical MTES for subsystems at level $i - 1$*
4. *Use MTS and diagnosability specification to decide which residual generators to implement*
5. *Derive residual generators for local MTES*
   Output:
   (a) *local residual generators for subsystem $M_{i,j}$*
6. *Derive residual generators for hierarchical MTES*
   Output:
   (a) *hierarchical residual generators for subordinate local diagnosers at level $i - 1$*

3.4.2 Decentralized diagnoser implementation

The diagnoser implementation consists of the following steps. The diagnosis process is assumed to start at the lowest level at which residuals sensitive to a fault exist. Higher layers of the duagnoser are contacted to isolate faults. The precise diagnoser implementation and process will be illustrated when applying the architecture to the ADCS in section 5.
The practical issues related to implementing such a diagnoser must be mentioned here. On a satellite, subsystems would be connected to a system bus. For calculation of hierarchical residual generators, values communicated by different subsystems are used. The delay suffered by these communicated values on the system bus would need to be taken into account. However these issues are out of the scope of this paper.

4. Attitude determination and control system of a LEO satellite

The structure of the attitude determination and control system of a typical satellite is represented in figure 5. The attitude determination subsystem (ADS) is composed of sensors which sense the rate and angular position of the satellite. An attitude estimate is achieved using sensor fusion, which is provided as input to the attitude control subsystem (ACS). The ACS is composed of the control signal calculation and the actuators which provide the stabilizing and/or control torque to the satellite. The satellite under study is assumed to be a three-axis stabilized satellite in orbit around the earth. We consider reaction wheels and magnetorquer as actuators. The modelling of the ADCS both for diagnoser design and simulation is summarized below.

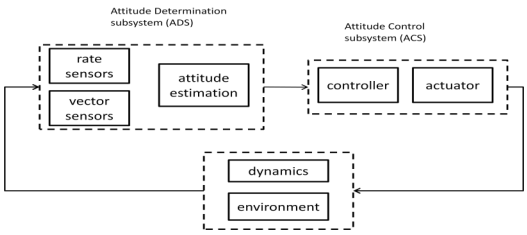

Figure 5: The ADCS of a typical satellite

4.1 Satellite Dynamics

The basic dynamic equations of a satellite motion can be summarized [9], [16] as:

$$I \cdot (\dot{\omega}) = T - (\omega \times (I \cdot \omega)) \tag{1}$$

$$T = T_d + T_m - T_w = [T_x, T_y, T_z] \tag{2}$$

Here T is the total torque acting along the body axes, while T_m, T_w and T_d are the torques vectors due to the magnetorquer, reaction wheels and disturbances respectively. The moment of inertia of the satellite body is represented as I, while ω is the angular velocity vector relative to an inertial frame.

4.2 ADCS modelling

The sensor suite of the satellite is composed of rate gyros for each of the three axes, and vector sensors which are used to periodically clear the accumulated attitude drift error from the rate gyroscopes. Sun and star sensors are examples of vector sensors. The development of the ADS follows that in [17] and [9]. The vector and rate sensor ouputs are used to estimate the state vector both independently and merged together. These preliminary estimates are then fused together to arrive at the estimate which is fed back to the ACS. These independent estimates provide an important redundancy in the ADS which can be used to check consistency.

The state vector of the satellite $\mathcal{X}$ is composed of the attitude angles pitch (θ), roll (ϕ) and yaw (ψ) and the corresponding rates i.e. $\mathcal{X} = [\psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}]$.

The ACS is composed of a reaction wheel assembly and magnetotorquers for momentum dumping. Additional sensors and actuators can be easily added to this structural model.

4.3 Fault scenarios

The structural model of the system is enriched with information about interesting faults. Following the development in [3], faults are introduced as signals in the system model equations. We consider faults on the rate and vector sensors of the ADS and the reaction wheels of the ACS. Such a fault class includes hard, soft and intermittent faults. The faults considered are summarized in Table 1. Each of the faults can have three components corresponding to the three axes.

Table 1: Fault scenarios of the ADCS

Component	Subsystem	Fault
Vector sensors (vs)	ADS	$fvs(fvs_x, fvs_y, fvs_z)$
Rate sensors (rs)	ADS	$frs(frs_x, frs_y, frs_z)$
Reaction wheel (rw)	ACS	$frw(frw_x, frw_y, frw_z)$

4.4 Structural modeling of the ADCS

The structure of the ADCS is abstracted as a set of constraints relating a set of variables. A discussion of such modeling, only for the ADS, can be found in [18]. The constraints are denoted by C in the following discussion. The constraints and variables involved are summarized in the tables that follow.

Most of the constraints C are composed of three behavioural relations corresponding to the three axes. The decomposition of the ADCS structure into the ADS and ACS subsystems is illustrated on the figure 6. These structures form the input for our decentralized architecture and algorithms. While the constraints and variables representing the dynamics of the satellite are listed separately, we consider them part of the ADS in section 5.

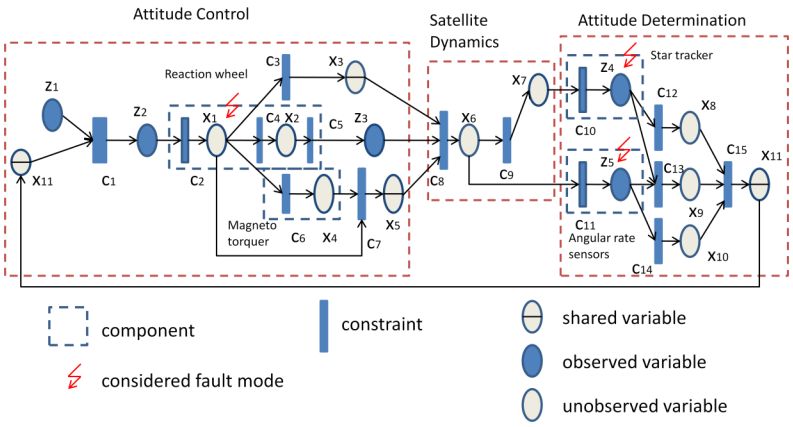

Figure 6: Structural modeling of the ADCS

From the set of variables of the system, the sensed quantities form the set of observed variables, with all the rest assumed to be unobserved. Some of the unobserved variables are internal states of the system whose value is available only through sensors. However others like state estimations are calculated quantities and can be available for diagnosis. The general procedure for diagnoser design starts with assuming a small set of observed quantities, which is expanded to fulfill dignosability and isolability specifications if required.

The X^{shared} set is composed of unobserved variables which propagate between the ADS and ACS subsystems.
$X^{shared} = \{T_{total}, h_w, \mathcal{X}_{est}\}$

Figure 7 shows the graphical representation of the ADCS structure with unobserved, observed and fault variables separated along the X-axis. The equations along the Y-axis are the behavioural relations of the system.

The constraints and variables that define the system are a composition of the constraints and variables of the subsystems. The structural models of the ADCS, ADS and ACS are represented as $(C_{ADCS}, X_{ADCS}, Z_{ADCS})$, $(C_{ADS}, X_{ADS}, Z_{ADS})$ and $(C_{ADC}, X_{ADC}, Z_{ADC})$ respectively. The structural model of the ADCS is composed of 42 equations in total with 42 unobserved variables, 15 observed variables and 9 faults which are modeled as variables in the equations.

Table 2: Constraints of the ADCS

Constraints	**Subsystem**	**Description**
$C_{control}/C_1$	ACS	Control algorithm
C_{RW1}/C_2	ACS	Reaction wheel motor dynamics
C_{RW2}/C_4	ACS	Reaction wheel flywheel dynamics
C_{RW3}/C_3	ACS	Reaction wheel angular momentum integration
C_{MT}/C_6	ACS	Magnetotorquer dynamics
$C_{summing}/C_7$	ACS	Total torque
$C_{tachometer}/C_5$	ACS	Tachometer
C_{dyn}/C_8	DYN (ADS)	Satellite dynamic equations of motion
C_{kin}/C_9	DYN (ADS)	Satellite kinematic equations of motion
C_{RS}/C_{11}	ADS	Rate sensors
C_{VS}/C_{10}	ADS	Vector sensors
C_{est1}/C_{12}	ADS	State estimation with vector sensor alone
C_{est2}/C_{13}	ADS	State estimation with both rate and vector sensors
C_{est3}/C_{14}	ADS	State estimation with rate sensors alone
C_{fusion}/C_{15}	ADS	Sensor fusion

Table 3: Unobserved variables of the ADCS

Unobserved Variable	**Subsystem**	**Description**
$\dot{h}_w/x_1$	ACS	Derivative of flywheel angular momentum
h_w/x_3	ACS	Flywheel angular momentum
ω_w/x_2	ACS	Flywheel angular speed
T_m/x_4	ACS	Magnetic torque
T_{total}/x_5	ACS	Total torque on satellite
$\mathcal{X}_{\omega}/x_6$	DYN (ADS)	Satellite angular rates
$\mathcal{X}_{pos}/x_7$	DYN (ADS)	Satellite attitude angles
$\mathcal{X}_{est1}/x_8$	ADS	Estimated satellite state with vector sensors alone
$\mathcal{X}_{est2}/x_9$	ADS	Estimated satellite state with rate and vector sensors
$\mathcal{X}_{est3}/x_{10}$	ADS	Estimated satellite state with rate sensors
$\mathcal{X}_{est}/x_{11}$	ADS	Estimated satellite state

Table 4: Observed variables of the ADCS

Observed Variable	**Subsystem**	**Description**
$\mathcal{X}_{ref}/z_1$	ACS	Reference value of state vector
T_c/z_2	ACS	Reaction wheel control torques
$\hat{\omega}_w/z_3$	ACS	Sensed value of reaction wheel flywheel angular speed
$\hat{\mathcal{X}}_{\omega}/z_5$	ADS	Sensed satellite angular rates
$\hat{\mathcal{X}}_{pos}/z_4$	ADS	Sensed satellite attitude angles

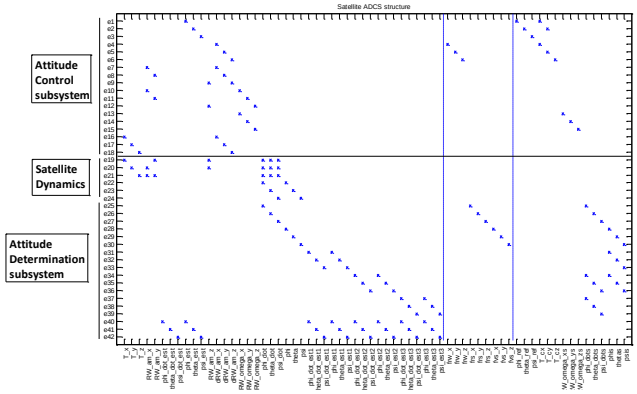

Figure 7: The ADCS structure of a typical satellite, and its decomposition into ACS and ADS

$$C_{ADCS} = C_{ACS} \cup C_{DYN} \cup C_{ADS} \tag{3}$$
$$X_{ADCS} = X_{ACS} \cup X_{DYN} \cup X_{ADS} \tag{4}$$
$$Z_{ADCS} = Z_{ACS} \cup Z_{DYN} \cup Z_{ADS} \tag{5}$$

The structural model of the ADCS described here will be used to demonstrate the proposed decentralized architecture in the next section, and to show that a global diagnoser design procedure is equivalent to the proposed decentralized procedure, i.e. the same MTES - and hence ARRs are obtained.

5. Diagnoser design for an Attitude Determination and Control System

Before applying the proposed decentralized diagnosis architecture to the ADCS, we demonstrate its need. Consider the results below of deriving MTES and MTS sets for the ADCS considered globally. Recall that the set of MTS represents the maximum fault isolability possible for a given structural description of a system. Corresponding to each MTS, the MTES represents the set of behavioural relations which theoretically could be used to derive residual generators sensitive to the faults in the MTS.

First we use the algorithm to derive MTES and MTS sets to the ADCS considered globally.

ADCS global diagnoser
Maximum fault isolability (MTS) : $[frw_x], [frw_y], [frw_z], [frs_x], [frs_y], [frs_z], [fvs_x], [fvs_y], [fvs_z]$
MTES : $[e4, e7, e10, e13], [e5, e8, e11, e14], [e6, e9, e12, e15], [e7 \ldots e21, e25], [e7 \ldots e21, e26], [e7 \ldots e21, e27], [e7 \ldots e21, e22, e28],$
$[e7 \ldots e21, e23, e29], [e7 \ldots e21, e24, e30]$
Number of MSO sets : 2448

These results for a centralized diagnoser will be used for comparison with the decentralized configuration below. Now we use the algorithm to derive the MTES and MTS sets for the ACS and ADS. Here the X^{shared} variables are unobserved; this information about whether X^{shared} is observed or not would be available globally, not locally.

ACS local diagnoser taking X^{shared} to be unobserved
Maximum fault isolability : $[frw_x], [frw_y], [frw_z]$
MTES : $[e4, e7, e10, e13], [e5, e8, e11, e14], [e6, e9, e12, e15]$

ADS local diagnoser taking X^{shared} to be unobserved
Maximum fault isolability : $[frs_x, fvs_x], [frs_y, fvs_y], [frs_z, fvs_z]$
MTES : $[e22, e25, e28], [e23, e26, e29], [e24, e27, e20]$

The results demonstrate that all the considered faults can be isolated with a centralized global diagnoser for the ADCS. All ACS faults can also be isolated by a local diagnoser. All ADS faults can be detected by its local diagnoser. However, faults on the rate and vector sensors cannot be isolated locally by a diagnoser working with the ADS local model. The high number of MSO sets which exist for the ADCS (2448) compared to the number of MTES (9) illustrates the massive computational advantage of only deriving MTES sets which correspond to the set of interesting faults, rather than all possible MSO sets.

The proposed decentralized architecture will now be applied to the ADCS by designing the local & supervisory diagnosers. It will be demonstrated that the isolability capability of such a decentralized diagnoser is equivalent to the global diagnoser above.

From the point of view of the local diagnosers, the shared variables X^{shared} are now assumed to be observed. They are assumed to be observed, as the local subsystem model does not have information about whether they are sensed in the other subsystems or if shared relations exist so that they can be expressed in terms of sensed variables. Shared MTES can be derived using the algorithm with this assumption.

ADS local diagnoser considering X^{shared} observed
Maximum fault isolability : $[frs_x], [frs_y], [frs_z], [fvs_x], [fvs_y], [fvs_z]$
Shared MTES : $[e19, e20, e21, e25], [e19, e20, e21, e26], [e19, e20, e21, e27], [e19, e20, e21, e22, e28], [e19, e20, e21, e23, e29], [e19, e20, e21, e24, e30]$

We see that complete fault isolability is achieved with our assumption. It is interesting to compare the MTES sets corresponding to faults which were not isolable before. We see the set $[e19, e20, e21]$ corresponding to the relations of C_{dyn} and relations from the set $[e22, e23, e24]$, i.e. the dynamic and kinematic equations of motion of the satellite respectively. C_{dyn} & C_{kin} are not functionally part of the ADS, even though these constraints are taken as part of the ADS in our implementation. Rather they are the interface between the ACS and the ADS and represent the physical behaviour of the satellite itself.

We note therefore that it might be possible to isolate (some of) the faults in the ambiguity sets $[frs_x \& fvs_x]$, $[frs_y \& fvs_y]$,$[frs_z \& fvs_z]$ if either some/all of the shared variables are sensed in the ACS, or shared relations exist in the ACS which allow these variables to be expressed in terms of observable variables.

ACS local diagnoser considering X^{shared} observed
Maximum fault isolability : $[frw_x], [frw_y], [frw_z]$
Shared MTES : $[e1 \dots e18], [e1 \dots e18], [e1 \dots e18]$

We still retain complete isolability, but the MTES sets [e1 - e18] representing the entire ACS structural model shows that there are various redundant ways of deriving the ARRs now. This is logical as observing X^{shared} would add more possibilites of deriving consistency checks. It should be noted that in practice we would derive residual generators for the ACS using the MTES sets $[e4, e7, e10, e13], [e5, e8, e11, e14], [e6, e9, e12, e15]$ as these ensure complete isolabilty of the ACS faults.

Lets use the shared MTES at the global level to derive the hierarchical MTES. We give this example for the faults in $[frs_x \ \& \ fvs_x]$

ADCS supervisory diagnoser to disambiguate faults
Input behavioural relations : $[e19, e20, e21, e22, e25, e28]$ & $[e1 - e18]$
Interesting fault vector : $[frw_x, frw_y, frw_z, frs_x, fvs_x]$
Maximum fault isolability : $[frw_x], [frw_y], [frw_z], [frs_x], [fvs_x]$
Hierarchical MTES : $[e4, e7, e10, e13], [e5, e8, e11, e14], [e6, e9, e12, e15], [e7 \dots e21, e25], [e7 \dots e21, e22, e28]$

The faults frs_x and fvs_x can be isolated now. The derived hierarchical MTES sets corresponding to these faults is exactly as that derived for a centralized ADCS diagnoser $[e7 \dots e21, e25]$ & $[e7 \dots e21, e22, e28]$. Similar results are obtained for the ambiguities corresponding to the other two fault ambiguity sets $[frs_y \ \& \ fvs_y]$ and $[frs_z \ \& \ fvs_z]$.

The functioning of the decentralized ADCS diagnoser is illustrated in figure 8. The local diagnosers run their local residual generator banks. Lets say a fault appears in the vector sensor suite of the ADS. The local diagnoser detects a fault, but cannot isolate it. So, a fault isolation request is sent to the supervisory level diagnoser, and the local diagnoser starts sending the relevant calculated shared relations from the ADS. The fault code could be $[frs_x, fvs_y]$ for example, indicating the source of the ambiguity. The supervisory layer will put the satellite into safe mode and then

request the ACS local diagnoser to start providing the relevant calculated shared relations. The hierarchical residual generators are then evaluated at the supervisory level. The fault is isolated, or if higher intervention is required, the ADCS diagnoser contacts the central diagnoser of the satellite.

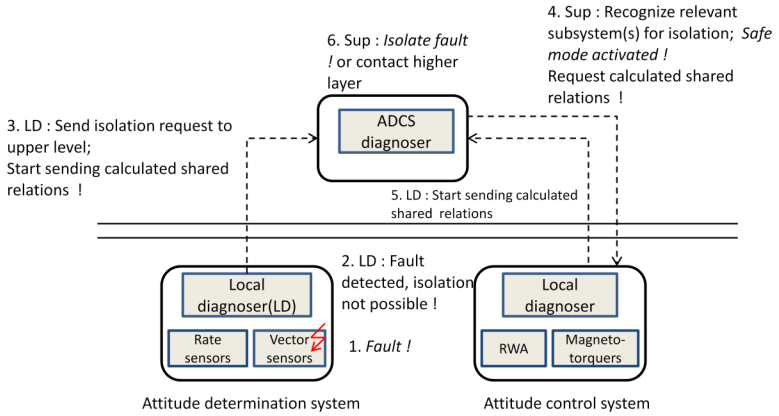

Figure 8: The decentralized diagnosis architecture and process applied to an ADCS

This is just one possible diagnoser functioning process possible with the architecture. Importantly, this process ensures firstly that only the smallest possible set of residual generators is evaluated during nominal operation, and secondly that communication bandwidth is not used under nominal operation for interaction between the local and supervisory diagnosers.

6. Conclusion

In this paper an architecture for decentralized fault diagnosis and isolation has been developed. Local diagnosers working with local models of their subsystems are coordinated by a supervisor at a higher level to resolve ambiguities arising out of quantities shared between subsystems. Isolation is performed by the supervisor on request. Practical and operational considerations will be kept in mind throughout during the design of the architecture. This framework is applied to the diagnosis of continuous systems using ARR based residual generators. Algorithms for the diagnosis of continuous systems are adapted and integrated into the architecture. The structural model of the attitude determination and control system (ADCS) of an earth orbiting satellite is developed in the paper and used to demonstrate our architecture. The ADCS is considered to be composed of the attitude determination and attitude control subsystems.

The designed residual generators will be implemented in the near future as residual generator banks, and the functioning of the architecture will be demonstrated. We are also proceeding to extend our framework so that the diagnosis of hybrid systems can be handled. We are focusing at the moment on [20] as an approach to hybrid system diagnosis. On the other hand to demonstrate the viability of such a decentralized architecture, we are developing a high fidelity spacecraft simulator with detailed models of some ADCS components, for example reaction wheels as in [6]. Realistic hard and soft faults will be modeled. Coupled with realistic disturbance profiles and noise such a testbed would serve the needs of the model based diagnosis and fault tolerant control communities.

Acknowledgement

The work described in this paper was carried out as part of a doctoral project jointly funded by the Centre National d'Études Spatiales(CNES), Toulouse and Thales Alenia Space. The authors wish to thank Marie-Claire Charmeau and Raymond Soumagne of CNES and Xavier Olive at TAS for their support and valuable comments.

References

[1] J. Kurien and M.D. R-Moreno, Costs and benefits of model-based diagnosis, In *Proccedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2008.

[2] M. Krysander, J. Åslund, and M. Nyberg, An efficient algorithm for finding minimal over-constrained subsystems for model-based diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1), 2008.

[3] M. Krysander, J. Åslund, and E. Frisk, A structural algorithm for finding testable sub-models and multiple fault isolability analysis, In *Proceedings of the 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, Oregon, October 2010

[4] C. Svard and M. Nyberg, Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems, *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 40(6):1310–1328, 2010

[5] N. Tudoroiu and K. Khorasani, Satellite fault diagnosis using a bank of interacting kalman filters, *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1334–1350, 2007

[6] B. Bialke, High fidelity mathematical modeling of reaction wheel performance, *Advances in the Astronautical Sciences, Guidance and Control*, 98:483–496, 1998

[7] M. S. Siva, N. Venkateswaran and P. S. Goel, Analytical redundancy based fault detection of gyroscopes in spacecraft applications, *Acta Astronautica*, 50(9):535–545, 2002

[8] A. Falcoz, D. Henry, and A. Zolghadri, Robust fault diagnosis for atmospheric reentry vehicles: a case study, *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 40:886–899, September 2010

[9] F. Sassani, F.N. Pirmoradi and C.W. de Silva, Fault detection and diagnosis in a spacecraft attitude determination system, *Acta Astronautica*, 65:710–729, 2009

[10] G. Gaydadjiev, N. E. Cornejo, R. Amini, Model-based fault detection for the Delfi-N3xt attitude determination system, In *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2010

[11] S. Simani, R.J. Patton, F. J. Uppal and B. Polle, Robust FDI applied to thruster faults of a satellite system, *Control Engineering Practice*, 18(9):1093–1109, 2010

[12] C. Picardi, L. Console, and D. T. Dupre, A framework for decentralized qualitative model based diagnosis, In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007

[13] L. Travé-Massuyès, T. Escobet, and Xavier Olive, Diagnosability analysis based on component-supported analytical redundancy relations, *Transactions on Systems, Man, and Cybernetics – Part A*, 36:1146–1160, 2006

[14] J. Armengol, A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès, Minimal structurally overdetermined sets for residual generation: A comparison of alternative approaches, In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 1480–1485, 2009

[15] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and Fault-Tolerant Control* Springer, 2003

[16] I. Zuliana and V. Renuganth, A study of reaction wheel configurations for a 3-axis satellite attitude control, *Advances in Space Research*, 45(6):750 – 759, 2010

[17] Marcel J. Sidi, *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, 1997

[18] H. Niemann, T. Lorentzen, and M. Blanke, Structural analysis - A case study of the Rømer Satellite, Automation, Ørsted, Technical University of Denmark

[19] E. Chanthery, S. Indra, and L. Travé-Massuyès, *The equivalence of global and decentralised ARRs computation*, Report LAAS No. N°11094, March 2011,

[20] M. Bayoudh and L. Travé-Massuyès, Hybrid systems diagnosis by coupling continuous and discrete event techniques, In *Proceedings of the 17th International Federation of Automatic Control, World Congress*, pages 7265 – 7270, 2008