Fault Isolation on Request Based on Decentralized Residual Generation

Elodie Chanthery, Louise Travé-Massuyès, Saurabh Indra

Fault Isolation on Request Based on Decentralized Residual Generation

E. Chanthery, L. Travé-Massuyès, and S. Indra,

***Abstract*—This paper presents the theoretical keystone for a decentralization of Model-Based Diagnosis by proving the equivalence between decentralized and centralized residual generation. The proof is based on structural analysis and graph-theoretical concepts. The second contribution of the paper is the design of a decentralized fault-focused residual generation scheme advantageously implementing a strategy of fault isolation on request. Algorithms are tested on the Attitude Determination and Control System of a Low Earth Orbit satellite.**

***Index Terms*—Decentralized residual generation, Model-Based Diagnosis, Structural Analysis, Focused Residual Generation, Isolation on Request**

I. Introduction

MODEL based diagnosis (MBD) detects and isolates faults based on a model of the system that can be developed either during the design phase or by a reverse engineering process.

Although interest in MBD is now well-established, there is a wide gap between the "state of art" and the "state of practice". This is often due to the extremely conservative nature of technology decisions and operations, as for example in the space domain. The main difficulty arises from the trade-offs between costs, benefits and risks associated with on-board MBD [1]. Increasing the applicability of MBD requires the development and adaptation of algorithms and architectures while keeping in mind the constraints and needs specific to the application field.

A way to ease the integration of MBD into real systems is to approach them as a set of several subsystems. As a consequence, while some subsystems follow conventional diagnosis methods, others may evolve and integrate new MBD techniques. Another important aspect is that industry is not open to sharing complete information about their systems, even more so if the development involves several firms. Decentralized solutions allow proper separation of the industrial knowledge, provided that inputs and outputs are clearly defined. Finally, a decentralized diagnosis architecture offers what is termed here a *fault isolation on request* capability while maintaining the same isolability power as centralized diagnosis.

The first contribution of the paper is to provide the formal proof of the equivalence of decentralized and centralized residual generation using the structural analysis approach [2]. The entire theoretical proof, which was sketched out in our previous work [3], [4], is provided. The structural model of a system is an abstraction of its behavioral model: only the structure of the constraints, i.e. the existence of links between variables, is considered, and not the constraints themselves [5].

The second contribution of the paper is the design, in a decentralized architecture, of a fault-focused residual generation scheme thanks to the notion of Minimal Test Equation Support [6]. We present a revision of the algorithm that was proposed in [3], [4] and provide details based on the equivalence proof that were omitted in previous papers. The algorithms are tested on the Attitude Determination and Control System (ADCS) of a spacecraft.

This paper goes beyond the work of the two previous papers [3] and [4] as it consolidates the theoretical foundations of the equivalence property and presents more realistic experiments to validate the proposed decentralized approach. It provides a comprehensive framework for decentralized model-based diagnosis using the analytical redundancy approach.

The paper is organized in the following way. Section II clarifies the notion of decentralization and motivates the work with respect to related work. Section III presents the background theory and prerequisites about structural analysis. The decentralized diagnosis problem is presented in Section IV. Section V demonstrates the equivalence of centralized and decentralized residual generation. Then Section VI examines the fault-focused residual generation problem. Section VII illustrates the theory by comparing the different residual generation algorithms on the ADCS case study. The paper is concluded in Section VIII.

II. Decentralized diagnosis: related work and motivations

This section first positions the decentralized diagnosis approach as compared to centralized and distributed approaches. It then discusses related work to motivate the contributions.

A. Motivations for decentralized diagnosis architectures

Three main categories of diagnosis architectures exist in the literature: centralized, decentralized and distributed (cf. Figure 1).

Definition 1 (Centralized diagnosis architecture): A *centralized* diagnosis architecture is composed of local agents without processing capabilities, typically sensors, that send data to a centralized diagnoser which computes the (global) diagnosis.

E. Chanthery L. Travé-Massuyès and S. Indra, are with CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

E. Chanthery is with Univ de Toulouse, INSA de Toulouse, LAAS, F-31400 Toulouse, France

L. Travé-Massuyès and S. Indra are with Univ de Toulouse, LAAS, F-31400 Toulouse, France

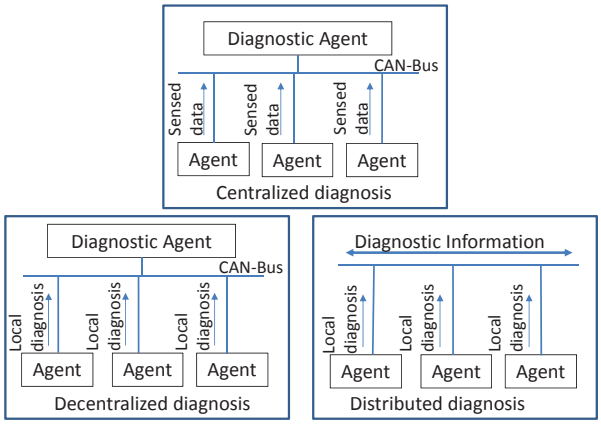

Fig. 1. Three architectures for diagnosis

The advantage of a centralized diagnostic system is its simplicity. There is no communication between local agents in the architecture. The main drawback is that it requires to explicitly building a global model of the system [7], which is unrealistic for large systems and when privacy issues come into play. Distributing and decentralizing diagnosis are two solutions to cope with these problems [8], [9].

Definition 2 (Distributed diagnosis architecture): A *distributed* diagnosis architecture assumes a set of local diagnosers, identical in terms of role, with communication possible between any two of them. A local diagnoser is an independent software entity. While local diagnosers are diagnoses for a subsystem, global diagnoses are diagnoses for the complete distributed system. Communication links must be designed so that local diagnoses are globally consistent [10].

Definition 3 (Decentralized diagnosis architecture): A *decentralized* diagnosis architecture is composed of local diagnosers whose results are coordinated by a supervisory diagnoser. There is no intra-level communication in the architecture. Inter-level communication between local diagnosers and the hierarchically upstream supervisory diagnoser serves to disambiguate local diagnosis results.

Decentralized and centralized diagnosis architectures differ in the place where processing is performed. In the centralized case, local agents are passive and they just collect and send data. Conversely, decentralized local agents compute local diagnoses and send them up to their supervisory diagnoser. In a distributed diagnosis architecture, local diagnoses are possibly shared by local agents without referring to any supervisory or central unit. This latter solution is scalable and robust. However, for domains like spacecrafts, this solution is not often investigated because it implies intense and advanced communication means. One solution, recently proposed by [9], is to define subsystems guided by the existing redundancies so that no communication is required. Local diagnosers are hence globally consistent by design. However the decomposition proposed in [9] ignores pre-existing constraints, that may be functional, geographical or privacy-based. We adopt an inverse approach and consider pre-existing constraints mandatory.

Most of the decentralized diagnosis methods deal with discrete event systems [7], [11], [12]. The decentralized diagnosis scheme of [7] is based on the "divide and conquer" principle and does not require computing a global model. We have the same motivations for continuous systems. [11] uses a decentralized approach to deal with the size of the model and to get a tractable representation of diagnoses. With the same idea, [12] proposes a hierarchical framework that capitalizes on local diagnoses. These ideas are also used in our approach with the notion of *isolation on request*. If local diagnosis is not sufficient, hierarchical diagnosers come into play to refine the diagnosis. Only in this case is hierarchical processing required, hence saving significant CPU time for on-line applications.

Decentralized diagnosis methods have been proposed only recently for continuous or hybrid systems. [13] presents a decentralized architecture for systems modeled in a qualitative framework. The architecture is quite similar to ours. However, diagnosis reasoning there relies on the logical model-based diagnosis theory whereas our approach adopts an analytical redundancy framework [14]. Addressing the fault tolerant control problem in a networked framework, [15] analyses fault detectability and fault isolability conditions for centralized, distributed and decentralized systems within a structural analysis framework. Nevertheless, the considered systems are linear. Our paper extends these results to nonlinear systems.

B. Structure of the decentralized diagnosis

In the proposed architecture, local diagnosers rely on models of their subsystems to arrive at diagnosis. Ambiguities might arise as faults propagate between subsystems. A supervisor at the higher level serves to resolve ambiguities and to provide diagnosis at a higher resolution than that possible with purely local information. The architecture is hierarchically scalable as can be seen in Figure 2.

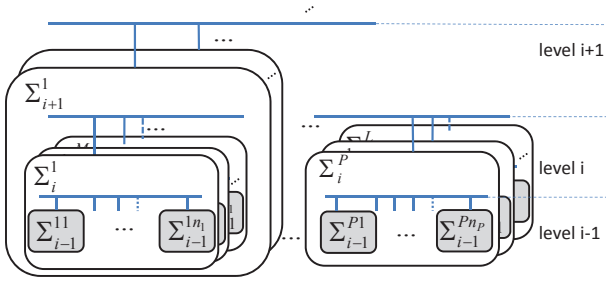

Fig. 2. Generic decentralized diagnosis structure

The decentralization levels are defined by the communication possibilities between diagnosers. Diagnosers of a level i communicate with their supervisory diagnoser of level $i+1$ and the diagnosers at the level $i-1$ below them in the hierarchy. We aim to expose as little information as possible about the subsystems.

III. Background Theory and Prerequisites

Our approach to diagnosis relies on *residual generators* designed from the *structural redundancies* of the system under

the following assumptions: 1) The system is component-based: a fault f_i corresponds to a component c_i and $f_i = 0$ if and only if c_i is not faulty; 2) Faults do not introduce "genuine" new equations and do not change the model structure; 3) The set of observable variables is known and fixed, i.e. no additional measurements become available once the diagnosis is initiated.

A. Structural approach to analytical redundancy

Analytical redundancy makes use of a model of the system to augment the number of tests that can be implemented [16]. These tests take the form of Analytical Redundancy Relations (ARRs) derived from the model and the knowledge of which variables are measured. The structural approach to deriving ARRs can be viewed as one of finding complete matchings on the bipartite graph of the system's model structural abstraction [5]. Even though the graph paradigm may not be the most efficient from a computational point of view [6], it provides a well-grounded theoretical framework to study properties.

1) Residual generators: Residual generators are derived based on ARRs [17] which are relations that involve only the measured variables of the system and their derivatives. A residual generator takes as input the values of the measured variables and, in an ideal case, gives a non-zero output only in case the system behavior is inconsistent with the model. Most of this development follows that in [6], [18] and [19].

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

Definition 4 (Model): A *model*, denoted $M(z, x)$ or M for short, is any set of equations relating z and x. The equations $r_i(z, x) \subseteq M(z, x)$, $i = 1, \ldots, n$, are assumed to be differential or algebraic in z and x.

The following model M is used to illustrate the concepts. It is composed of six equations r_1 to r_6 relating the unknown variables $X = \{x_1, x_2, x_3, x_4, x_5\}$ and the known variables $Z = \{u, v, w, y\}$.

$$\left.\begin{array}{lrcl} r_1: & \dot{x_1} & = & -x_1^2 + x_3 + u \\ r_2: & \dot{x_2} & = & x_4^2 \\ r_3: & x_1 & = & 3x_2^3 + v \\ r_4: & y & = & x_5 \\ r_5: & \ddot{x_3} & = & x_4^2 + x_5 \\ r_6: & x_3 & = & w - x_5 \end{array}\right\} M \qquad (1)$$

Definition 5 (Consistency): A model $M(z, x)$ is said to be consistent with a given trajectory of z, or concisely, consistent with z, if there exists a trajectory of x such that the equations $M(z, x)$ are fulfilled.

Definition 6 (ARR for $M(z, x)$): 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.

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

Definition 7 (Residual Generator for $M(z, x)$): 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 $\lim_{t\to\infty} r(t) = 0$.

2) Structural modeling and ARR generation: The *structure* of the system can be abstracted as a representation of which variables are involved in the equations that make up the model of the system. This abstraction leads to a bipartite graph $G(M \cup X \cup Z, A)$, or equivalently to $G(M \cup X, \mathcal{A})$, where $\mathcal{A} \subseteq A$ and $\mathcal{A}$ is a set of edges such that $a(i, j) \in \mathcal{A}$ iff variable x_i is involved in relation r_j. As shown in Figure 3, the bipartite graph (on the right) may be equivalently represented as a biadjacency matrix (on the left). Known variables u, v, y and w are not represented.

	unknown				
	x_1	x_2	x_3	x_4	x_5
r_1	×		×		
r_2		×		×	
r_3	×	×			
r_4					×
r_5			×	×	×
r_6			×		×

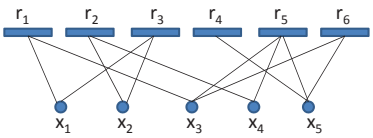

Fig. 3. Structural abstraction of M: biadjacency matrix and bipartite graph

The structural abstraction allows us to identify structural redundancies in the form of *Structural ARRs (SARRs)* [19], known as causal interpretations of *Minimal Structurally Overdetermined sets* (MSO set) [6] or *Possible Conflicts* [20], independently of the linear or nonlinear nature of the systems. However one must keep 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 used for the design of residual generators.

It can be shown [2] that ARRs can be derived from so-called complete matchings between X and M on the bipartite graph $G(M \cup X, \mathcal{A})$.

Definition 8 (Matching): A *matching* between X and M is a subset of $\mathcal{A}$ such that no vertex in $X \cup M$ is incident with more than one edge of the matching. A matching is complete if it covers every vertex of X.

A complete matching between X and M is denoted by $\mathcal{M}(X, M)$, or $\mathcal{M}$ when there is no ambiguity. $\mathcal{M}(X, M)$ provides a way to identify the paths to calculate the unknown variables from the measured variables.

Figure 4 illustrates a complete matching for M indicated by circled entries in the biadjacency matrix (left) or bold edges in the corresponding bipartite graph (right). For instance, the unknown variable x_3 is matched to the relation r_1. Relation r_6 is *redundant* because it is not involved in the complete matching. This means that r_6 is not needed to calculate the unknown variables and that it can be used to check for consistency.

The substitution of unknown variables into a redundant equation allows one to obtain an ARR, as can be seen in the bottom part of Figure 4 that shows the substitution path.

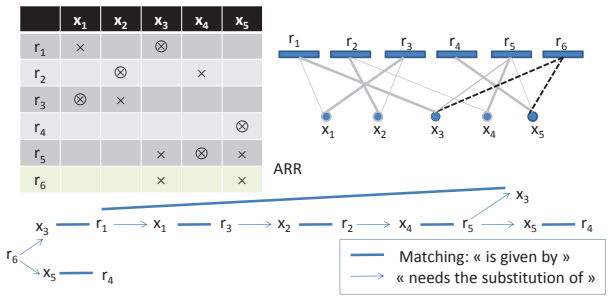

Fig. 4. Structural derivation of a SARR

The substitution path gives the ordered dependences between relations and variables. The *structural redundancy* of M is one, hence one redundant relation is available to derive an ARR.

B. Graph theory background

We now introduce the notions of graph theory needed to prove the equivalence of decentralized and centralized residual generation.

Definition 9 ($\mathcal{M}-(un)saturated$ vertex): Given a matching $\mathcal{M}$ in a bipartite graph $G(M \cup X, \mathcal{A})$, a vertex v is said to be $\mathcal{M}-(un)saturated$ if there is an (no) edge of $\mathcal{M}$ incident with v.

Definition 10 ($\mathcal{M}$-alternating path): Given a matching $\mathcal{M}$ in a bipartite graph $G(M \cup X, \mathcal{A})$, an *$\mathcal{M}$-alternating path* is a path in $G(M \cup X, \mathcal{A})$ whose edges are alternately members and non members of $\mathcal{M}$.

Definition 11 (Path length): The length of a path composed of edges and vertices in a matching $\mathcal{M}$ is the number of vertices in the path.

Definition 12 (Substitution path): A substitution path is an $\mathcal{M}$-alternating path of odd length that begins with a vertex in M.

For the matching of Figure 4, vertices r_1, r_2, x_1, x_2 are $\mathcal{M}-saturated$ while vertex r_6 is $\mathcal{M}-unsaturated$. The path x_1, r_3, x_2 is an $\mathcal{M}$-alternating path. The path r_1, x_1, r_3 is a substitution path whereas x_1, r_3, x_2 is not.

Property 1: The substitution of a variable x_i in a relation r_j involves exactly one edge (x_i, r_k) in the matching, and one edge (r_j, x_i) outside the matching. The path (r_j, x_i, r_k) is said to be a *minimal $\mathcal{M}$-alternating path*.

Proof: Assume that a variable x_i in a relation r_j can be substituted using a relation r_k. This implies that the edge (r_j, x_i) is not part of the matching, and that r_k is matched to x_i. ■

In the example, x_1 in r_1 can be substituted using r_3. (x_1, r_1) is not in the matching, (x_1, r_3) is in the matching. The path (r_1, x_1, r_3) is a minimal $\mathcal{M}$-alternating path.

The necessary and sufficient conditions for the existence of a complete matching on a bipartite graph are provided by *Hall's Theorem* [21].

Theorem 1 (Hall's Theorem): Consider a bipartite graph $G(M \cup X, A)$, where $|X| \leq |M|$. Then G has a complete matching $\mathcal{M}$ saturating every vertex of X if and only if $|S| \leq |N(S)|$ for every subset $S \subseteq X$, where $N(S)$ is the subset of vertices of M that are adjacent to some vertices in S i.e $N(S) = \{v \in M : \exists u \in S; (u, v) \in A\}$.

Property 2: The existence of a substitution path between two vertices x and r, is a sufficient condition for the existence of a complete matching for the subgraph that is defined by the substitution path.

Proof: By definition, a substitution path P is an $\mathcal{M}$-alternating path of odd length that begins with a vertex in M. Suppose that $P = \{r_1, x_1, r_2, x_2, \ldots, r_n, x_n\}$ and $G'(\{r_1, \ldots, r_n, x_1, \ldots, x_n\}, A')$, where A' is the set of edges in P is the subgraph defined by P. If (r_1, x_1) is in $\mathcal{M}$ then (x_1, r_2) is not in $\mathcal{M}$. By recurrence, it is possible to prove that for all $i = 1, \ldots, n, x_i$ is saturated. If (r_1, x_1) is not in $\mathcal{M}$ then (x_1, r_2) is in $\mathcal{M}$. By recurrence, it is possible to prove that for all $i = 1, \ldots, n, x_i$ is saturated. In conclusion, as for all $i = 1, \ldots, n, x_i$ is saturated, then $\mathcal{M}$ is a complete matching for $G'(\{r_1, \ldots, r_n, x_1, \ldots, x_n\}, A')$. ■

IV. Decentralized Diagnosis: problem formulation

A. Decentralization and related notions

In the following, "global" means no decentralization and, without loss of generality, only two hierarchical levels, so-called local level and supervisory level, are considered.

A decomposition of the system M, with associated bipartite graph $G(M \cup X \cup Z, A)$, into several subsystems M_i is defined as a partition of its equations.

Formally, let $M = \{M_1, M_2, \ldots, M_n\}$ with $M_i \subseteq M$, $M_i \neq \varnothing$, $\bigcup_{i=1}^{n} M_i = M$, and $M_i \cap M_j = \varnothing$ if $i \neq j$.

Definition 13 (Variables of a subsystem i): Given a bipartite graph $G(M \cup X \cup Z, A)$, the set of variables of the i^{th} subsystem, denoted as X_i and Z_i, are defined as the subset of vertices of X and Z respectively that are adjacent to some vertices corresponding to variables in M_i:

$$\begin{aligned} X_i &= \{u \in X : \exists v \in M_i, (u, v) \in A\} \\ Z_i &= \{u \in Z : \exists v \in M_i, (u, v) \in A\}. \end{aligned}$$

This decomposition 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 14 (Local variables): The set of *local variables* of the i^{th} subsystem, denoted as X_i^{local}, is defined as the subset of vertices of X_i that are adjacent only to vertices in M_i, and not to vertices in any other subsystem M_j, $j \neq i$,

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

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

Definition 15 (Shared variables): The *shared variables* X^{shared} are defined as:

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

Figure 5 illustrates the decomposition of M into two subsystems Σ_1 and Σ_2, with $M_1 = \{r_1, r_2, r_3\}$ and $M_2 = \{r_4, r_5, r_6\}$. We have $X_1^{local} = \{x_1, x_2\}$, $X_2^{local} = \{x_5\}$ and $X^{shared} = \{x_3, x_4\}$ as illustrated in the bottom part of the figure. The circled entries should be ignored for now.

Definition 16 (Local complete matching): A *local complete matching* $\mathcal{M}_i$ for the i^{th} subsystem is a complete matching between local variables X_i^{local} and local behavioral equations M_i in the bipartite graph $G(M_i \cup X_i^{local}, \mathcal{A}_i)$, where $\mathcal{A}_i \subset A_i$ is the set of edges incident to vertices of X_i^{local} and M_i.

Definition 17 (Global complete matching): A *global complete matching* $\mathcal{M}$ is a complete matching between X and M on the bipartite graph $G(M \cup X, \mathcal{A})$.

Figure 5 illustrates a global complete matching for M (top), and local complete matchings for Σ_1 and Σ_2 (bottom). A circle indicates a relation and its matched unknown variable.

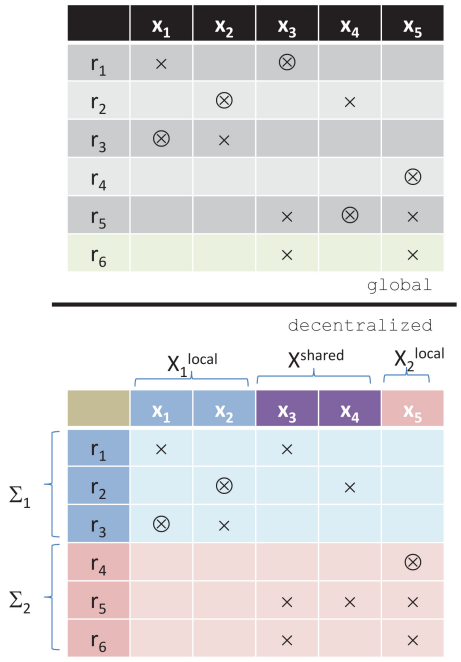

Fig. 5. From a global system to a decentralized system

Property 3: Given a global complete matching $\mathcal{M}$ for a system, let $M^{matched}$ be the set of vertices that are matched, and $M^{available} = M \backslash M^{matched}$. For each relation in $M^{available}$, it is possible to substitute every unknown variable using $\mathcal{M}$. So each behavioral relation r in $M^{available}$ potentially leads to an ARR.

Definition 18 (Hierarchical and Source Relations): Let us consider the local bipartite graphs $G(M_i \cup X_i^{local}, \mathcal{A}_i), i = 1, \ldots, n$, and suppose that a local complete matching $\mathcal{M}_i$ exists for each of them. Consider also the set of relations that are not matched in any local complete matching $\mathcal{M}_i$, i.e. the *locally redundant relations*. Let r be one of these relations. By construction, r relates a set of variables, among which unknown variables belong to only one of the X_i^{local} and to X^{shared}. Using $\mathcal{M}_i$, it is possible to substitute in r each variable belonging to X_i^{local}, and so obtain a new relation r^h involving only unknown variables in X^{shared}. The new relation r^h is called a *hierarchical relation* because it refers to several subsystems, and hence belongs to the supervisory level. The relation r is called the *source relation* of r^h. The set of hierarchical relations is denoted as R^h.

A locally redundant relation r may involve only variables of X^{shared}, in which case it is called a *pure hierarchical relation*. On the other hand, r may involve only variables of X_i^{local} but result in a hierarchical relation that involves variables of X^{shared} after the substitution process.

Definition 19 (Hierarchical complete matching): A hierarchical complete matching $\mathcal{M}^h$ is a complete matching between the shared variables X^{shared} and the hierarchical relations R^h in the bipartite graph representing the structure of the system at the supervisory level i.e. $G^h(R^h \cup X^{shared}, \mathcal{A}^h)$.

Figure 6 shows the hierarchical relations r_1^h, r_5^h, and r_6^h resulting from the source relations r_1, r_5, and r_6, respectively, and the local complete matchings shown in Figure 5 (bottom). The substituted variables in the hierarchical relations are indicated by *hash* symbols.

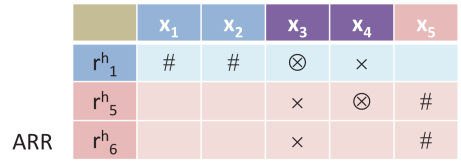

Fig. 6. Structural derivation of hierarchical redundant relations

Property 4: Assume that r is a source relation and r^h the associated hierarchical relation. An edge (r^h, x) between the hierarchical relation and a shared variable x exists at the hierarchical level if either the edge (r, x) is present in the bipartite graph of the global system or there exists at least one substitution path that links x to r at the global level.

Proof:

- If the edge (r, x) exists for the global bipartite graph, then a corresponding edge (r^h, x) obviously exists at the hierarchical level.
- If the adjacency (r, x) does not exist at the global level, this implies that the edge (r^h, x) derives from a finite number of variable substitutions in r. Lets say that k substitutions are needed. According to Proposition 1, the substitution of x requires exactly two additional edges, so k substitutions would involve $2k$ additional edges. Consequently, the path between r^h and x involves $2k+1$ edges that alternately belong to $\mathcal{M}$. By definition, such a path is a substitution path. ■

V. The Equivalence of Centralized and Decentralized Diagnosis

When designing decentralized diagnosers for a system, it is desirable that properties such as fault detectability and isolability are not altered by decentralization. This can be ensured if the set of ARRs derived with the global and decentralized architectures are identical. This section formalizes this equivalence and provides the proof.

Proposition 1: Assume that the system M is decomposed into subsystems $M_1, M_2, \dots M_n$, then the set of ARRs that can be derived for M with a centralized architecture is identical to the set of ARRs that can be derived for the decentralized system by deriving the ARRs for each subsystems $M_i, i = 1, \dots, n$, and for the hierarchical relations.

The proof of this proposition relies on proving that the *set of behavioral relations* involved in a complete matching at the global level, and the ones used to derive a set of local and hierarchical complete matchings *are the same*. It follows that the set of ARRs which can be derived are the same.

A. From global to local: "If" proof

Proposition 2: Given a global complete matching $\mathcal{M}$ in the bipartite graph $G(M \cup X, \mathcal{A})$ that leads to a non-void set of ARRs, then for any decomposition $\{M_1, M_2, \cdots, M_n\}$ of the system M, there exists a set of local complete matchings $\{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_n\}$ and a hierarchical complete matching $\mathcal{M}^h$ that result in the same set of ARRs.

The principle of the proof is as follows. When the system is decomposed into subsystems, each relation matched with a shared variable in $\mathcal{M}$ can now be used as the source relation for a hierarchical relation. And, at the hierarchical level, a shared variable can be matched with the corresponding hierarchical relation. Consequently, the local matchings $\{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_n\}$, and the hierarchical matching $\mathcal{M}^h$ involve the same set of relations as the global complete matching $\mathcal{M}$ and therefore result in the same ARRs.

Proof: Given the existence of $\mathcal{M}$, Hall's theorem indicates that $|X| \leq |M| + 1$. Without loss of generality, consider only the set of matched relations $M_{matched}$. Then, $|M_{matched}| = |X|$.

(1) In each of the subsystems $M_i, i = 1 \dots n$, the matches of the original global matching $\mathcal{M}$ can be preserved when they match local variables $x_i \in X_i^{local}$. These matches constitute the local complete matching for the local bipartite graph $G(M_i \cup X_i^{local}, \mathcal{A}_i)$.

(2) Each shared variable x in X^{shared} is matched to one relation r_x in $M_{matched}$. As a consequence of (1), the relation r_x is not involved in a local matching. Such relation can be used to derive a hierarchical relation r_x^h, which can still be matched to x. This set of matches is hence a hierarchical complete matching.

Summarizing, for each variable x in X, *either* x is a local variable, i.e. $x \in X_i^{local}$, and can be matched with the same relation in the local matching $\mathcal{M}_i$ as in the global matching $\mathcal{M}$, *or* x is a shared variable, i.e. $x \in X_{shared}$, and can then be matched to the hierarchical relation r_x^h constructed from the relation r_x to which it was matched in the global matching $\mathcal{M}$.

As the matching relations are exactly the same, the decentralized and centralized diagnoser design process lead to the same ARRs from a structural perspective. ■

B. From local to global: "Only if" proof

Proposition 3: Assume that the system M is decomposed into subsystems $M_1, M_2, \dots M_n$. Given $\{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_n\}$ a set of local complete matchings in the bipartite graphs $G(M_i \cup X_i^{local}, \mathcal{A}_i)$, $i = 1 \dots n$ and $\mathcal{M}^h$ the hierarchical complete matching in the bipartite graph $G^h(R^h \cup X_{shared}, \mathcal{A}_i)$, there exists a global complete matching $\mathcal{M}$ in $G(M \cup X, \mathcal{A})$ that results in the same set of ARRs.

The principle of the proof is as follows. A hierarchical complete matching implies the existence of either a global complete matching, i.e. on $G(M \cup X, \mathcal{A})$, or of a set of substitution paths in the subsystems which permit 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 those involved in the global complete matching, therefore leading to the same set of ARRs.

Proof: First, consider the hierarchical complete matching $\mathcal{M}^h$. The hierarchical relation that initially belongs to subsystem $G(M_i \cup X_i^{local}, \mathcal{A}_i)$ and that is matched with x in $\mathcal{M}^h$ is denoted by $r_x^{i^h}$. Due to *Property 4*, there are two possible cases for the existence of each hierarchical match $\mathcal{M}^h$:

(1) the match corresponds to an edge (r, x) that already exists at the global level, and the match hence remains the same,

(2) there exists at least one substitution path that links the shared variable x to the source relation r_x at the global level. Due to *Property 2*, there exists a complete matching for the subgraph defined by the substitution path. This subgraph is totally included in the bipartite graph of subsystem $G(M_i \cup X_i^{local}, \mathcal{A}_i)$ by the construction process of the hierarchical relation $r_x^{i^h}$.

Consequently, for each of the local complete matchings $\mathcal{M}_1, \dots, \mathcal{M}_n$, two possibilities exist. Either they are preserved in the global matching $\mathcal{M}$, or if not, the same set of relations are involved. In other words, the set of relations involved in the global complete matching is exactly the same as the ones that are involved in the local matchings and the hierarchical matching. Hence, the global complete matching leads to the same set of ARRs. ■

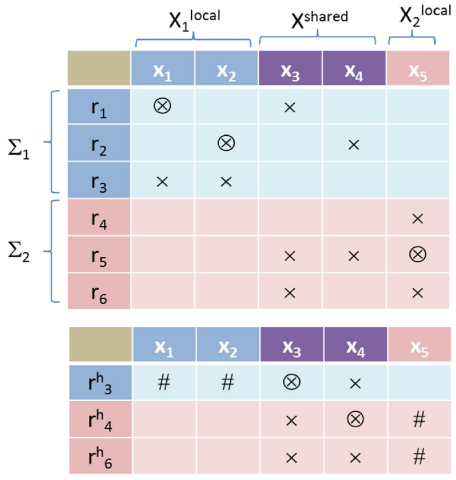

Fig. 7. Local complete matchings are not preserved

These arguments can be illustrated using the example of Figures 4, 6 and 7. Figure 6 illustrates the situation when the decentralized local matchings are preserved from the global

case given in Figure 4. On the other hand, in Figure 7 (top), the local complete matching for the first subsystem is different from the one corresponding to the global matching of Figure 4, although hierarchical relations (bottom) are the same.

VI. Focused Residual Generation

It is possible to analyze the structural properties of a system modeled as a set of equations M involving variables X by using the Dulmage-Mendelson (DM) canonical decomposition as shown in Figure 8.

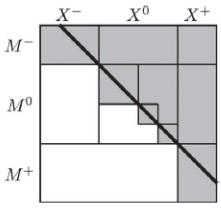

Fig. 8. Dulmage-Mendelson decomposition of a model M [18]

This decomposition results in the partition 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 underdetermined part represented by M^-. The sets defined below formalize the notions of a structurally overdetermined equation set (SO) and a proper structurally overdetermined (PSO) equation set [6].

Definition 20 (SO): A set M of equations is structurally overdetermined (SO) if M has more equations than unknowns.

Definition 21 (PSO): 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 [6], namely the MSO sets, are of special interest since they are at the core of the isolability properties.

Definition 22 (MSO): An SO set is a minimal structurally overdetermined (MSO) set if no proper subset is an SO set.

ARRs correspond to MSO sets, which are sets of equations with one more equation than unknowns [6]. Unknown variables can be solved using the set of equations minus one, and then the last one is a redundant equation that can be used to check for consistency. We adopt an MSO set based ARR design method for our decentralized diagnoser architecture.

A. Fault-focused structural ARR generation

An efficient algorithm to compute all possible MSO sets for a system is developed in [6]. However only MSO sets corresponding to relevant faults are interesting to construct residual generators, hence a *fault-focused* procedure. [18] introduces the concept of Test Equation Support (TES), which is a set of equations expressing redundancy specific to a set of considered faults. This set of faults is known as the *Test Support* (TS). A minimal TES (MTES) and a minimal TS (MTS) are such that no proper subset is a TES and TS, respectively.

An MSO set or an MTES circumscribes the presence of structural redundancy that can be used to check consistency for a part of the system. It is interesting to notice that, whereas an MSO gives rise to one single residual generator, an MTES gathers all the equations leading to all the possible residual generators for the associated set of faults. MTES are hence a more compact way than MSOs to identify residual generators. 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 [6]. To generate residuals from MTESs, we use the method proposed in [22]. It relies on developing a computational sequence to successively solve for the unknown variables involved in a redundant 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 9. The offline structural analysis and diagnoser design is illustrated at the top of the figure. The input of the diagnoser design is the structural model M, including information about interesting faults $F(M)$. The MTES and associated MTS are then computed using the algorithm of [18] (box "*MTES algorithm for finding testable sub models*"). The residual generator method is then used (box "*Calculating analytical residual generators*") for the MTES selected by the *diagnosability specification* module. The analytical residual generators are stored in the "*residual generator bank*" that is used on-line, fed by system inputs and outputs as illustrated at the bottom of Figure 9. Fault isolation is carried out after fault detection using fault signatures which are vectors composed of the binary residual bank output (0/non0) resulting from an appropriate statistical test.

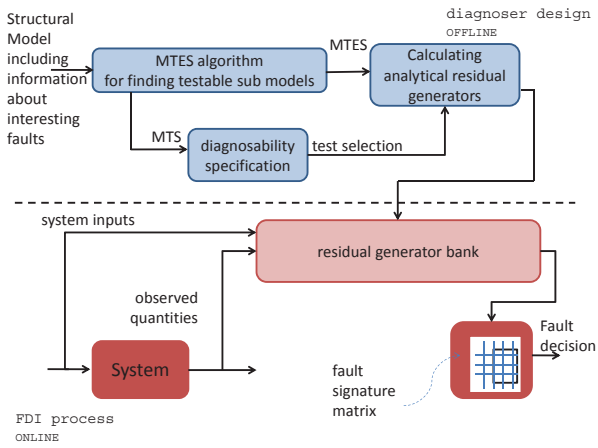

Fig. 9. The design and implementation scheme for a centralized global diagnoser

B. Decentralized diagnoser design

Figure 10 can be considered as the decentralized counterpart of Figure 9. It represents the diagnoser for a subsystem at level i and shows the communication required between diagnoser levels.

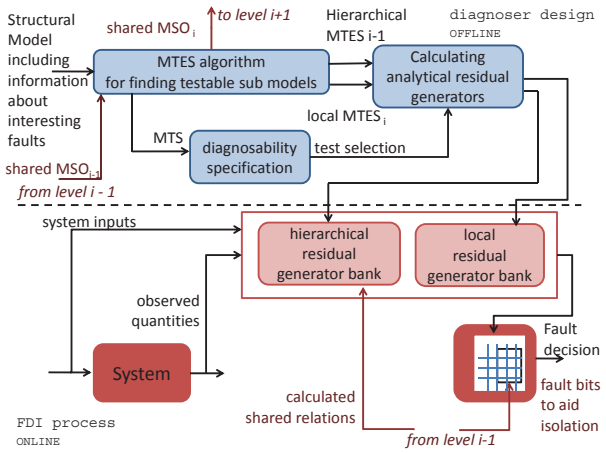

Fig. 10. The design and implementation scheme of a decentralized diagnoser for a subsystem at level i

Local TES/MTES are defined as containing only local variables. They otherwise contain shared variables and are called *shared TES/MTES*.

The diagnoser design is done offline and consists of the steps below, performed for each subsystem $M_{i,j}$ $j = 1 \dots n_i$ at each level $i = 1 \dots n_l$, in a nested loop. i is the level in the hierarchy, and j the enumeration of subsystems at that level:

1) *Use the MTES algorithm with the structural model of the subsystem $M_{i,j}$ as input at level i, considering shared variables as known (box "MTES algorithm for finding testable submodels")*
   Output: *local MTESs for the subsystem $M_{i,j}$* at level i; *MTS for the subsystem $M_{i,j}$* at level i; *shared MSOs for the subsystem $M_{i,j}$* at level i to be sent at level $i+1$ (arrow "*to level i+1*").
2) *Use the MTES algorithm with the shared MSOs coming from subordinate local diagnosers of level $i-1$*
   Output: *hierarchical MTESs for subsystems at level $i-1$.*
3) *Use MTS and diagnosability specification to decide which residual generators to implement (box "diagnosability specification")*
4) *Derive residual generators from local MTES*
   Output: *local residual generators for subsystem $M_{i,j}$ stored in the "local residual generator bank".*
5) *Derive residual generators from hierarchical MTESs*
   Output: *hierarchical residual generators for subordinate local diagnosers of level $i-1$ stored in the "hierarchical residual generator bank".*

The aim of step 2) is to replay the MTES algorithm to eliminate shared variables (that are assumed known locally). It may happen that these variables are just involved in local redundancies without fault support, i.e. are involved in an MSO but not an MTES. This is why shared MSOs are needed at step 2). After the offline structural analysis, the diagnoser is implemented as a residual generator bank that is used on-line, fed by system inputs and outputs as illustrated at the bottom of Figure 10).

C. Extension of the equivalence for MTESs

The goal of this section is to extend the equivalence of centralized and decentralized diagnosis to the use of Test Equation Supports (TESs).

Proposition 4: The set of MTESs computed in a centralized way and the set of MTESs computed in a decentralized way ends to the same set of ARRs.

Proposition 4 is obvious from Property 1 and the fact that the diagnoser design steps compute the set of hierarchical MTES from the set of shared MSOs.

VII. Application to the Attitude Determination and Control System of a Satellite

Our work has been tested on the Attitude Determination and Control System (ADCS) of a Low Earth Orbit satellite. After a short description of the system and the fault scenarios, we derive MTES and TES sets for the ADCS in a centralized way. Then we compare this set to MTES and TES sets obtained with a decentralized architecture. The last part of this section describes the functioning of the decentralized ADCS diagnoser. The centralized and decentralized ARR based methods are compared in terms of net benefits and on simulated scenarios.

A. The ADCS system

1) Satellite Dynamics: The basic dynamic equations of satellite motion can be summarized as [23], [24]:

$$I \cdot (\dot{\omega}) = T - (\omega \times (I \cdot \omega)) \quad (2)$$

$$T = T_d + T_m - T_w = [T_x, T_y, T_z] \quad (3)$$

Here T is the total torque acting along the body axes, while T_m, T_w and T_d are the torque 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.

2) ADS and ACS modeling: 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 [25] and [23]. The vector and rate sensor outputs 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.

B. Structural modeling of the ADCS

1) Fault scenarios: The structural model of the system is enriched with information about interesting faults. Following the development in [18], 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 as illustrated in Figure 11 with a broken arrow. The considered fault types include hard, soft and intermittent faults. The faults considered are summarized in Table I. Each of the faults can have three components corresponding to the three axes.

TABLE I
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)

2) Structural modeling: The structure of the ADCS is abstracted as a set of relations $R = \{r_i\}$ relating sets of unknown and known variables $X = \{x_i\}$ and $Z = \{z_i\}$. It is represented in Figure 11 in which the variables known/unknown or local/shared involved in every relation appear explicitly. For example, the relation r_1 is linked to z_1 and z_2 (known variables) and x_{11} (shared variable).

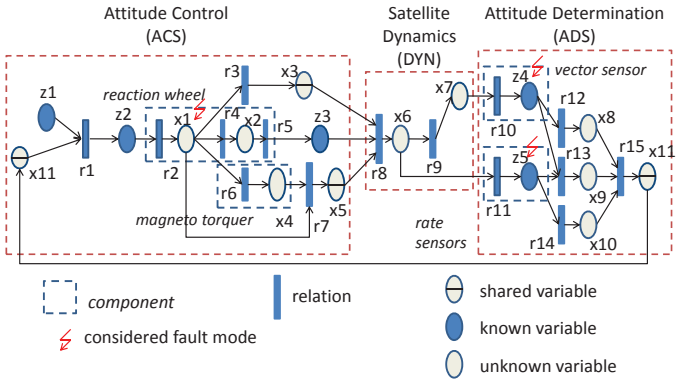

Fig. 11. Structural modeling of the ADCS

A discussion of such modeling, only for the ADS, can be found in [26]. The relations and variables involved are summarized in the Tables II, III and IV. Most of the relations are composed of three behavioral equations corresponding to the three axes indexed by subscripts x, y, and z.

While the relations and variables representing the dynamics of the satellite (DYN) are listed separately, we consider them part of the ADS in Section VII-C. These relations form the input for our decentralized architecture and algorithm.

The X^{shared} set is composed of unknown variables which propagate between the ADS and ACS subsystems: $X^{shared} = \{T_{total}(x_5), h_w(x_3), \mathcal{X}_{est}(x_{11})\}$ where $T_{total} = [T_x, T_y, T_z]^{\mathrm{T}}$, and $h_w = [RW_{am_x}, RW_{am_y}, RW_{am_z}]^{\mathrm{T}}$, $\mathcal{X}_{est} = [\psi_{est}, \theta_{est}, \phi_{est}, \dot{\psi}_{est}, \dot{\theta}_{est}, \dot{\phi}_{est}]^{\mathrm{T}}$.

Figure 12 shows the biadjacency matrix of the ADCS structure with unknown, known and fault variables separated along the X-axis. The equations along the Y-axis are the behavioral equations of the system.

The structural model of the ADCS is composed of 42 equations in total with 42 unknown variables, 15 known variables and 9 faults modeled as variables in the equations.

TABLE II
RELATIONS OF THE ADCS

Relations	Subsystem	Description
$r_{control}/r_1$	ACS	Control algorithm
r_{RW1}/r_2	ACS	Reaction wheel motor dynamics
r_{RW2}/r_4	ACS	Reaction wheel flywheel dynamics
r_{RW3}/r_3	ACS	Reaction wheel angular momentum integration
r_{MT}/r_6	ACS	Magnetotorquer dynamics
$r_{summing}/r_7$	ACS	Total torque
$r_{tachometer}/r_5$	ACS	Tachometer
r_{dyn}/r_8	DYN (ADS)	Satellite dynamic equations of motion
r_{kin}/r_9	DYN (ADS)	Satellite kinematic equations of motion
r_{RS}/r_{11}	ADS	Rate sensors
r_{VS}/r_{10}	ADS	Vector sensors
r_{est1}/r_{12}	ADS	State estim. with vector sensor alone
r_{est2}/r_{13}	ADS	State estim. with both rate and vector sensors
r_{est3}/r_{14}	ADS	State estim. with rate sensors alone
r_{fusion}/r_{15}	ADS	Sensor fusion

TABLE III
UNKNOWN VARIABLES OF THE ADCS

Unknown Var.	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	Estim. sat. state with vector sensors alone
$\mathcal{X}_{est2}/x_9$	ADS	Estim. sat. state with rate and vector sensors
$\mathcal{X}_{est3}/x_{10}$	ADS	Estim. sat. state with rate sensors
$\mathcal{X}_{est}/x_{11}$	ADS	Estim.sat. state

C. MTES equivalence of centralized and decentralized diagnosers

1) ADCS centralized diagnoser: First we use the algorithm to derive MTES and MTS sets for the ADCS considered globally. Faults appearing in the same MTS are not discriminable. The list of equations e_i, $i = 1 \ldots n$, is denoted $e_1 \ldots e_n$.

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

The results demonstrate that all the considered faults can

TABLE IV
KNOWN VARIABLES OF THE ADCS

Known Var.	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

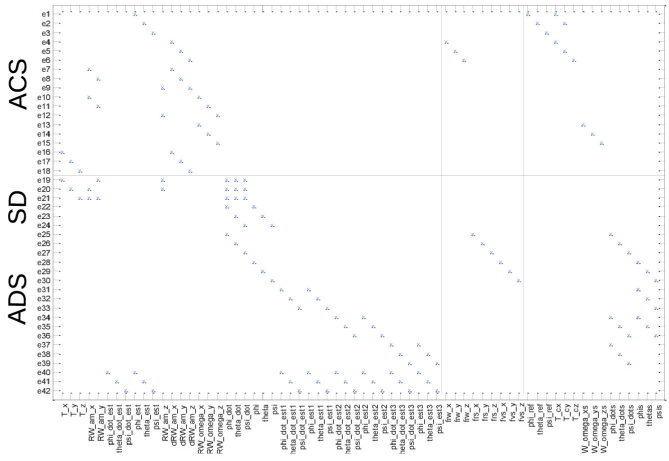

Fig. 12. The ADCS structure and its decomposition into ACS and ADS

be detected and isolated with a centralized diagnoser for the ADCS. The high number of MSO sets (2448) compared to the number of MTES (9) illustrates the great computational advantage of deriving MTES sets focused on the set of interesting faults. These results will be used for comparison with the decentralized configuration later on.

2) Achievable diagnosability based on local diagnosers only: We use the algorithm to derive the MTES and MTS sets for the ACS and ADS separately assuming that the shared variables X^{shared} are unknown.

ACS local diagnoser (taking X^{shared} to be unknown):
MTS: $[frw_x]$, $[frw_y]$, $[frw_z]$
Local MTES: $[e4, e7, e10, e13]$, $[e5, e8, e11, e14]$, $[e6, e9, e12, e15]$

ADS local diagnoser (taking X^{shared} to be unknown)
MTS: $[frs_x, fvs_x]$, $[frs_y, fvs_y]$, $[frs_z, fvs_z]$
Local MTES: $[e22, e25, e28]$, $[e23, e26, e29]$, $[e24, e27, e20]$

From the results, we can conclude that all ACS faults can be detected and isolated by the local ACD diagnoser. On the other hand, although all ADS faults can be detected by the local ADS diagnoser, faults on the rate and vector sensors cannot be isolated.

3) ADCS decentralized diagnosis architecture: The proposed decentralized architecture is now applied to the ADCS by designing the local and supervisory diagnosers.

From the point of view of the local diagnosers, the shared variables X^{shared} are now assumed to be known. Local MTES remain the same as derived in the previous subsection. Shared MTES are derived below using the new observability assumption.

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

Complete fault isolability is achieved with the current assumption. Let us notice that the shared MTES sets supported by the faults that were not isolable before include relations $e19$, $e20$, $e21$ and relations $e22$, $e23$, $e24$, i.e. the dynamic and kinematic equations of motion of the satellite C_{dyn} and C_{kin}, respectively. These are not functionally part of the ADS, even though they are taken as part of the ADS in our implementation. Rather they are the interface between the ACS and the ADS, representing the physical behavior of the satellite itself.

It can be concluded that it is 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 which allow these variables to be expressed in terms of known variables of the ACS.

ACS local diagnoser considering X^{shared} known
MTS: $[frw_x]$, $[frw_y]$, $[frw_z]$
Shared MTES: $[e1, e2, e3, e6 \dots e18]$, $[e1, e2, e3, e5, e7 \dots e18]$, $[e1, e2, e3, e4, e7 \dots e18]$

Obviously, shared MTES sets do not increase isolability since for the ACS, full isolability was already achieved with local MTES sets. Shared MTES sets do not bring any improvement from this point of view. Nevertheless, they have a degree of structural redundancy equal to 10 compared to 1 for ACS local MTES sets. There are hence various redundant ways of deriving the ARRs now. X^{shared} would indeed add possibilities of deriving consistency checks. In practice however, residual generators would be derived using the local MTES sets $[e4, e7, e10, e13]$, $[e5, e8, e11, e14]$, $[e6, e9, e12, e15]$ because they involve less equations and result in less complex consistency checks.

Let us use the shared MSOs of ACS and ADS at the global level to derive the hierarchical MTES sets. If we focus the search on the entire set of faults, we get the following results:

ADCS supervisory diagnoser to disambiguate faults:
Input behavioural relations: $[e1 \dots e18]$ and $[e19 \dots e30]$
Fault vector under focus: $[frw_x, frw_y, frw_z, frs_x, frs_y, frs_z, fvs_x, fvs_y, fvs_z]$
MTS: $[frw_x]$, $[frw_y]$, $[frw_z]$, $[frs_x]$, $[frs_y]$, $[frs_z]$, $[fvs_x]$, $[fvs_y]$, $[fvs_z]$
Hierarchical MTES: $[e4, e7, e10, e13]$, $[e5, e8, e11, e14]$, $[e6, e9, e12, e15]$, $[e7 \dots e21, e25]$, $[e7 \dots e21, e26]$, $[e7 \dots e21, e27]$, $[e7 \dots e21, e22, e28]$, $[e7 \dots e21, e23, e29]$, $[e7 \dots e21, e24, e30]$

Consistent with the equivalence result proved in Section V, the derived hierarchical MTES sets are exactly that derived for the centralized ADCS diagnoser.

More interestingly, if we aim at implementing an isolation on request architecture, we can focus the search on the ambiguity set $[frs_x, fvs_x]$:

ADCS supervisory diagnoser to disambiguate faults
Input behavioural relations: $[e1 \dots e18]$ and $[e19 \dots e30]$
Fault vector under focus: $[frs_x, fvs_x]$
MTS: $[frs_x], [fvs_x]$
Hierarchical MTES: $[e4 \dots e24, e26 \dots e30]$, $[e4 \dots e21, e23 \dots e27, e29, e30]$

Hierarchical MTES sets have a degree of structural redundancy equal to 8. The faults frs_x and fvs_x can be isolated now. Similar results are obtained for the ambiguities corresponding to the other two fault ambiguity sets $[frs_y, fvs_y]$ and $[frs_z, fvs_z]$.

D. The decentralized ADCS diagnoser in operation

The operation of the decentralized ADCS diagnoser is illustrated in Figure 13.

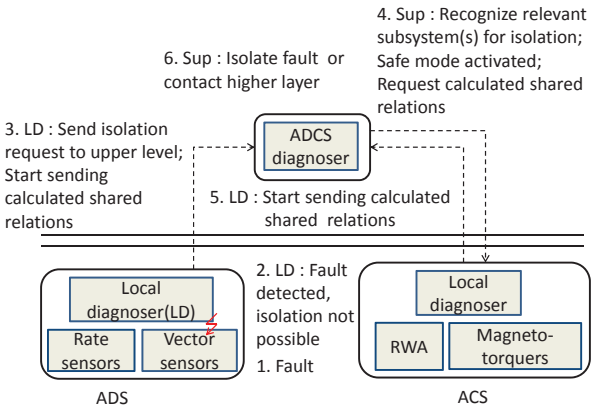

Fig. 13. The decentralized diagnosis architecture and process applied to an ADCS (the numbers 1 to 6 indicate the sequence of operations)

The local diagnosers run their local residual generator banks. Let us say a fault appears in the z-axis rate sensor. Figure 14 illustrates that the local diagnoser detects the fault as the ADS local residual for the z-axis is not equal to zero. It cannot isolate it because this residual reacts both to frs_z and fvs_z.

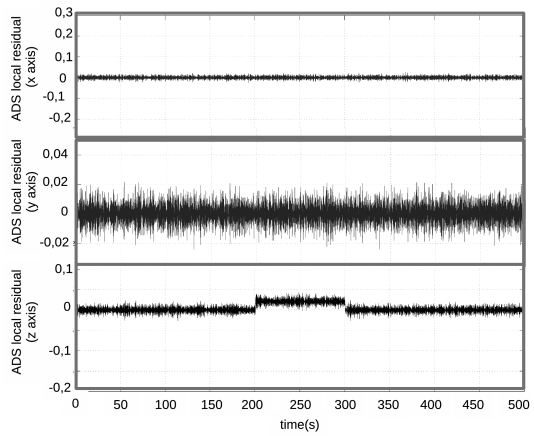

Fig. 14. ADS local residuals

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 is $[frs_z, fvs_z]$, indicating the source of the ambiguity. The hierarchical residual generators are then evaluated at the supervisory level as shown in Figure 15. Fault frs_z is successfully isolated.

Importantly, this process ensures firstly that only the smallest possible set of residual generators is evaluated during nominal operation, which implies better reactivity of the diagnoser, and secondly that communication bandwidth is not used under nominal operation for interaction between the local and supervisory diagnosers. The price to pay is a slower response of the diagnoser when hierarchical levels are involved. In this work, we prioritize fast response in nominal situations, which results in significant computational gain on average.

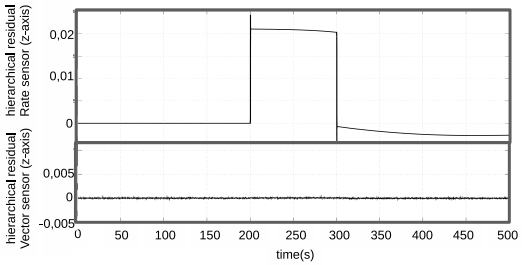

Fig. 15. ADS hierarchical residuals for the z-axis

E. Gauging the benefits of diagnoser decentralization

We have laid out in this paper a framework for decentralized ARR based diagnoser design. Such a groundwork enables the design and development of model-based diagnosis systems within the classical system enginering methodology. In this way diagnoser development practice can be brought closer to the development methodology of nominal functionality.

Instead of requiring full visibility, in such a decentralized system engineering approach: 1. the diagnoser design and algorithm is shared among project partners and an interface among diagnosers is enforced with different partners responsible for the various functional units; 2. an underlying system simulation is shared, with interface signals visible to all partners as required; 3. the partners provide diagnosers implemented first in a modeling language such as MATLAB/Simulink and then as the project proceeds in the target SW language to be validated together on the integration simulator.

The hierarchical architecture implies that the method can be scaled to much larger systems when the importance of such design and development considerations is critical.

The considerations involved in the development and functional verification of real-world FDIR systems in space applications can be found in [27]. Previous work has compared the design and development effort of the decentralized ARR based diagnoser with conventional rule based error monitors [28].

Studying the number of thresholds, differentiators and associated filtering ($Diff.$), and integrators and associated filtering ($Int.$) provides a simple way to gauge the change in cost-benefit analysis on decentralization. While setting simultaneous thresholds adds to tuning and validation effort, differentiators and integrators introduce computational and numerical complexity to the tuning, validation and run-time efforts. In actual applications it is these considerations that tip the balance in favor of the utilized technique (cf. Table V). Also, while these figures are for our two levels case study, which is small, the multiplicative effect on the figures as the size of the system increases is considerable.

VIII. Conclusion

This paper presents the theoretical keystone for a decentralization of model-based diagnosis within the analytical

TABLE V
COMPARISON BETWEEN CENTRALIZED AND DECENTRALIZED ARR BASED DIAGNOSERS

Diagnoser	$Thresholds$	$Diff.$	$Int.$
Centralized	9 per fault	9	3
Decentralized	6 per fault + 2 per fault (IoR*)	3+ (IoR) 2	3

*Isolation on Request

redundancy framework. The need for decentralized diagnosis is justified by many applications, such as spacecrafts whose architecture is organized into functional modules, and developed with decentralized systems engineering. The demonstration of the equivalence between decentralized and centralized approaches for residual generation is done using the structural approach. The second contribution is a fault focused decentralized residual generation design method. The algorithms have been implemented and tested on the ADCS of a satellite. This case study illustrates the advantages of the decentralized diagnosis architecture, which offers lower complexity and isolation on request capabilities while maintaining the same isolability power as centralized design.

Future work will focus on the optimization of the process underlying decentralization being considered as a design problem. Whereas we believe that the choice of subsystems is dictated by design constraints, functionality for monolithic systems or geographical location for distributed systems, the definition of the hierarchical layers and the selection of residual generators leaves space for optimization.

ACKNOWLEDGMENT

This work was supported by Thales Alenia Space, France and the French Space Agency, CNES.

REFERENCES

[1] J. Kurien and M. R-Moreno, "Costs and benefits of model-based diagnosis," in *Aerospace Conference, 2008 IEEE*, march 2008.

[2] M. Blanke, M. Kinnaert, and J. Lunze, *Diagnosis and Fault-Tolerant Control*. Springer, 2006.

[3] S. Indra, L. Trave-Massuyes, and E. Chanthery, "A decentralized FDI scheme for spacecraft: Bridging the gap between model based FDI research & practice," in *4th European Conference for Aerospace Sciences*, 2011.

[4] ——, "A decentralized fault detection and isolation scheme for spacecraft: bridging the gap between model-based fault detection and isolation research and practice," *Progress in Flight Dynamics, GNC, and Avionics*, vol. 6, pp. 281–298, 2013.

[5] J. P. Cassar and M. Staroswiecki, "A structural approach for the design of failure detection and identification systems," *IFAC Control of Industrial Systems*, 1997.

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

[7] Y. Pencolé and M. Cordier, "A formal framework for the decentralized diagnosis of large scale discrete event systems and its application to telecommunication networks," *Artificial Intelligence*, vol. 164(1-2), pp. 121–170, 2005.

[8] J. Biteus, M. Nyberg, and E. Frisk, "An algorithm for computing the diagnoses with minimal cardinality in a distributed system," *Engineering Applications of Artificial Intelligence*, vol. 21-2, pp. 269–276, 2008.

[9] A. Bregon, M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, and B. Pulido, "An event-based distributed diagnosis framework using structural model decomposition," *Artificial Intelligence*, vol. 210, pp. 1–35, 2014.

[10] P. K. John and A. Grastien., "Local consistency and junction tree for diagnosis of discrete-event systems," in *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, 2008, pp. 209–213.

[11] M.-O. Cordier and A. Grastien, "Exploiting independence in a decentralised and incremental approach of diagnosis," in *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[12] Y. Wang, T.-S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems*, vol. 17, no. 2, pp. 233–263, Jun. 2007.

[13] L. Console, C. Picardi, and D. T. Dupre, "A framework for decentralized qualitative model based diagnosis," in *20th International Joint Conference on Artificial Intelligence*, 2007.

[14] L. Travé-Massuyès, "Bridging control and artificial intelligence theories for diagnosis: A survey," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 1–16, 2014.

[15] D. Sauter, T. Boukhobza, and F. Hamelin, "Decentralized and autonomous design for FDI/FTC of networked control systems," in *Fault Detection, Supervision and Safety of Technical Processes*, vol. 6 Part 1, 2006, pp. 138–143.

[16] E. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *Automatic Control, IEEE Transactions on*, vol. 29, no. 7, pp. 603–614, 1984.

[17] 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*, 2009, pp. 1480–1485.

[18] M. Krysander, J. Åslund, and E. Frisk, "A structural algorithm for finding testable sub-models and multiple fault isolability analysis," in *Proceeding of the 21st International Workshop on Principles of Diagnosis (DX-10)*, 2010.

[19] L. Travé-Massuyès, T. Escobet, and X. Olive, "Diagnosability analysis based on component-supported analytical redundancy relations," *Trans. Sys. Man Cyber. Part A*, vol. 36, pp. 1146–1160, 2006.

[20] B. Pulido and C. A. González, "Possible conflicts: a compilation technique for consistency-based diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 5, pp. 2192–2206, 2004. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TSMCB.2004.835007

[21] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: Elsevier, 1976.

[22] C. Svard and M. Nyberg, "Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems," *Trans. Sys. Man Cyber. Part A*, vol. 40(6), pp. 1310–1328, 2010.

[23] F. Pirmoradi, F. Sassani, and C. de Silva, "Fault detection and diagnosis in a spacecraft attitude determination system," *Acta Astronautica*, vol. 65, pp. 710–729, 2009.

[24] I. Zuliana and V. Renuganth, "A study of reaction wheel configurations for a 3-axis satellite attitude control," *Advances in Space Research*, vol. 45, no. 6, pp. 750 – 759, 2010.

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

[26] T. Lorentzen, M. Blanke, and H. Niemann, "Structural analysis - a case study of the romer satellite," in *Fault detection, supervision and safety of technical processes symposium, SAFEPROCESS 2003*, 2003.

[27] D. Pecover, "Functional verification of the ADM-AEOLUS autonomy requirements," in *SpaceOps 2010 Conf.*, 2010.

[28] S. Indra and L. Travé-Massuyès, "Spacecraft fault detection and isolation system design using decentralized analytical redundancy," in *Advances in Aerospace Guidance, Navigation and Control*. Springer Berlin Heidelberg, 2013, pp. 247–263. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38253-6_16

Elodie Chanthery Elodie Chanthery received the engineering degree from "Ecole Nationale Supérieure dElectrotechnique, dElectronique, dInformatique, dHydraulique et des Tlecommunications" in 2002. She has defended her Ph.D degree, prepared at the "Ecole Nationale Supérieure de l'Aéronautique et de l'Espace-SupAero", in 2005 about mission planning for autonomous vehicles. She is an assistant professor in control at the Institut National des Sciences Appliquées (INSA) in Toulouse, France. Her research activities are conducted in LAAS-CNRS in the field of complex hybrid systems, focusing on the links between diagnosis and other tasks such as planning or prognosis.

Louise Travé-Massuyès Louise Travé-Massuyès is Research Director of the Centre National de la Recherche Scientifique (CNRS), leader of the "Diagnosis and Supervisory Control" (DISCO) Team in the LAAS-CNRS research laboratory, Toulouse, France. Her main research interests are in Dynamic Systems Supervision and Diagnosis with special focus on Qualitative, Model-Based Reasoning methods and data mining. She has been particularly active in bridging the AI and Control Engineering Model-Based Diagnosis communities, as leader of the BRIDGE Task Group of the MONET European Network. She has been responsible from several industrial and European projects and published more than 250 papers in scientific journals and international conference proceedings and 4 books. She is coordinator of the Maintenance & Diagnosis Strategic Field within the Aerospace Valley World Competitiveness Cluster, and serves as the contact evaluator for the projects submitted to the French Research Funding Agency. She serves in the Editorial Board of the Artificial Intelligence Journal. She is member of the IFAC Safeprocess Technical.

Saurabh Indra Saurabh Indra was born in New Delhi, India in December' 1983. He completed his masters degree at the Swiss Federal Institute of Technology in Lausanne (EPFL). Since October' 2009 he has been working on his doctoral degree at LAAS-CNRS, Toulouse in the field of model based diagnoser design for spacecraft. Since early' 2013, he has been working in the space industry in the areas of functional verification, and AOCS design and development.