Decentralized Diagnosis with Isolation on Request for Spacecraft ⋆

S. Indra[1,4,*], L. Travé-Massuyès[1,2], and E. Chanthery[1,3]

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2] Univ of Toulouse, LAAS , F-31400 Toulouse, France
[3] University of Toulouse, INSA , F-31400 Toulouse, France
[4] Centre National d'Études Spatiales, Toulouse
* *sindra@laas.fr, louise@laas.fr, echanthe@laas.fr*

Abstract This paper presents a decentralized diagnoser architecture for fault detection and isolation of continuous systems with a focus on space applications. The diagnosis algorithm utilized in the architecture is based on analytical redundancy relations. Local diagnosers work on functional subsystems with a supervisory diagnoser at the higher level responsible for resolving ambiguities arising from the interaction between subsystems. We demonstrate how the decentralized architecture can be used to address some of the key problems associated with spacecraft diagnoser design. These issues are the opacity of diagnoser structure, integration of diagnoser design and development into the system engineering framework, and the reduction of computation and communication overheads. Varying diagnosability levels can be realized depending on the mission phase. We develop diagnostic models of a satellite attitude determination and control system, and then design a decentralized diagnoser based on these models. Simulation results for a case study are presented.

Keywords: Decentralized diagnosis, Spacecraft diagnosis, Analytical redundancy relations, Fault protection systems

1 Introduction

Robust and effective fault diagnosis is an enabling technology for the ambitious spacecraft and missions of the future. The fault detection and isolation (FDI) schemes currently implemented onboard spacecraft are constrained by their reliance on rule based methods. These schemes map symptoms to possible diagnosis, suffer from opacity in design and behaviour and lead to decreased robustness.

Increasing the robustness of a spacecraft and its mission constitutes the system health management task. The subset of health management implemented onboard the spacecraft is known as fault protection (FP) Morgan (2011). Model based diagnosis (MBD) can serve a key role in spacecraft fault protection systems due to its ability to reason based on an underlying model of the system. However, there is a wide gulf between the MBD techniques applicable to spacecraft and the mission pull for utilizing these techniques onboard operational spacecraft. Among the various reasons for this gap are the computational complexity of most model based approaches, the extremely conservative nature of technology decisions and operations in the space domain and issues with the cost-benefit analysis Kurien and R-Moreno (2008). Increasing the applicability of MBD to spacecraft FP requires the development and adaptation of algorithms and architectures while keeping in mind the constraints and needs specific to the field. The work discussed in this paper serves as a step in this direction.

⋆ This work was sponsored by the Centre National d'Études Spatiales and Thales Alenia Space. We would like to thank Raymond Soumagne of CNES and Xavier Olive of TAS.

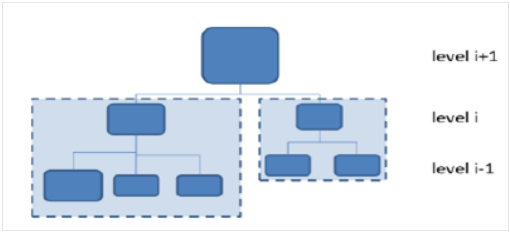

Figure 1. The Diagnoser Architecture

We discuss a decentralized, hierarchically scalable diagnosis architecture and illustrate its design and implementation for a satellite attitude determination and control system (ADCS). A representation of this architecture can be seen in figure 1. Local diagnosers detect and isolate faults in their subsystems utilizing local models. Ambiguities arise because of the quantities exchanged between subsystems. These ambiguities can be resolved at the higher layers of the decentralized architecture, based on the *isolation on request* (I_{req}) interface. As opposed to a distributed architecture where local diagnosers can communicate with each other, there is no communication among local diagnosers at one level in our architecture. We utilize the structural approach to modeling a system for deriving the structure of the analytical redundancy relations (ARRs). This paper extends S. Indra (2011) by discussing the decentralization of diagnosability capability and by demonstrating the isolation on request framework in simulation.

Distributed execution of the diagnoser is enabled by the architecture. There are, however important motivations for decentralized diagnoser development even in the case of a centralized implementation. The development of aerospace vehicles

usually relies on a systems engineering process, Tumer (2011). System level requirements are functionally decomposed for subsystems. There has been much recent interest in integrated design and development of a system and its health management capabilities. This is referred to as *function-failure codesign*. The proposed decentralized architecture and development process fits into such a framework. Mission stages determine diagnosability requirements for subsystems and components. The entire global diagnoser does not have to be executed continuously with the decentralized architecture. Sections of the diagnoser can be executed based on the diagnosability requirements in a certain phase of the mission. Such operation of the diagnoser keeps its communication and computational demands during nominal operation to a minimum.

The paper is structured as follows. The decentralized architecture and interface are discussed in section 2. The satellite ADCS models for diagnosis and simulation are presented in section 3. Diagnoser design and implementation is the subject of section 4. Section 5 concludes the paper with some perspective for future work.

2 The Decentralized Diagnosis Architecture

We utilize an ARR based approach to FDI within a structural framework Blanke et al. (2006). Redundancies present in parts of a system are used to check for consistency between sensed and expected quantities. A survey of algorithms to analyse the structure of a system detecting redundant portions for use in ARR based methods is provided in Armengol et al. (2009). A summary of the theoretical background to the diagnosis approach we use is discussed first in this section. Most of this discussion follows that in Krysander et al. (2008), Krysander et al. (2010) and Travé-Massuyès et al. (2006). Next we extend ARR based diagnosis algorithms for our decentralized architecture. Then the diagnoser architecture is described followed by the diagnoser design and implementation steps.

2.1 Background of diagnosis algorithms

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.

Definition 1. (ARR for $M(z, x)$Armengol et al. (2009)). 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 in Armengol et al. (2009). 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. Two formalisms can be used to represent this structure, bipartite graphs and adjacency matrices. Such an 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. It has been shown that ARRs correspond to minimal structurally over determined (MSO) sets, which are sets of equations of the system with one more equation than unknowns Armengol et al. (2009). 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 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.

An efficient algorithm to compute all possible MSO sets for a system is developed in Krysander et al. (2008). But the redundant equation sets which need to be exploited to construct residual generators can be limited to those which correspond to interesting faults. Krysander et al. (2010) 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 subsets expressing minimal redundancies are denoted minimal TES (MTES) and minimal TS (MTS).

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. This requirement was built in the algorithm proposed in Armengol et al. (2009) resulting in double exponential complexity. The residual generator derivation approach proposed in Svard and Nyberg (2010) 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 constitutes a sequential residual generator.

2.2 Notions for decentralized diagnosis

This section introduces the notions we need in order to devise the proposed decentralized architecture.

Hypothesis 1. A decomposition of a system M, with associated bipartite graph $G(M \cup X \cup Z, A)$, into several sub-systems 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 2. (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 subsystems leads to n subsystems denoted $M_i(x_i^{local}, z_i)$, with as-

sociated 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 3. (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 4. (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, $\forall i (1, \ldots, n), X_i^{local} \cap X^{shared} = \varnothing$.

Definition 5. (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 6. (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 7. (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 8. (Hierarchical complete matching). A hierarchical complete matching $\mathcal{M}_h$ is a complete matching between X^{shared} and R' on the graph $G_h(R' \cup X^{shared}, A')$.

2.3 *The equivalence of centralized and decentralized diagnosis*

We want to ensure 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.

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

2.3.1 *Sufficiency 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.

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

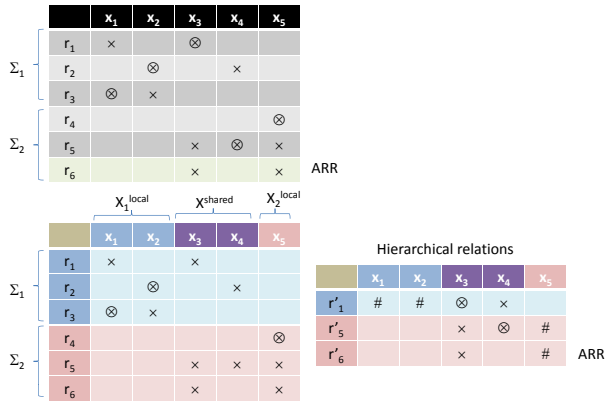

Figure 2. From a global system to a distributed system

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 2 shows the decomposition of a system into 2 subsystems, and the resulting matchings. It is important to note that we consider two subsystems in the figure to simplify the explanation. However, this is an oversimplification in the sense that it might not illustrate why a formal proof of equivalence is required. The situation becomes complicated as the number of subsystems increases, and as one subsytem might share variables with several others. For reasons of space only the proof idea is explained below. A discussion of a more general situation, and the complete proof can be found in S. Indra (2011).
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 2, 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 $\mathcal{M}$.

2.3.2 *Necessity 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.

2.4 *The Diagnoser Architecture and Development Process*

The diagnoser design and implementation steps of the diagnosis process can be seen in figure 3. We consider the diagnoser for a subsystem at level i of the diagnoser hierarchy of figure 1.

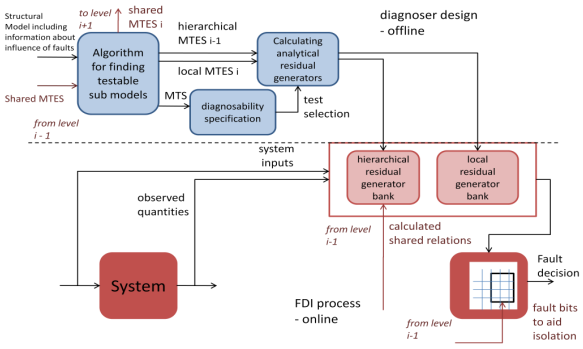

Figure 3. The design and implementation scheme of a decentralized diagnoser for a subsystem at level i

Corresponding to local and hierarchical complete matchings, there exist local and hierarchical MTES sets while shared MTES sets correspond to hierarchical relations. These are illustrated in figure 3. Utilizing the structural model of the system at level i, and the shared MTES from the lower levels, hierarchical, local and shared MTES for this level are derived. Diagnosability specifications guide the selection of residual generators to be implemented. The analytical expressions of the residual generators are then derived and implemented as hierarchical and local residual generator banks. Practical issues with distributed computation of residuals such as synchronization of communicated values will need to be dealt with, but are outside the scope of this paper.

As modelling costs are a significant component of the development costs of diagnosers, there is much current work on automating the development of diagnosis models. The structural models utilized for diagnoser development can be extracted from simulation and control models developed for the subsystem/system. Decisions about the framework of the model, the required granuality and sensor placement possibilities at a level are key issues. Looking at the development process from the perspective of a supplier-integrator relationship, the need for exposure of local models should be kept down to a minimum.

2.5 *Isolation on Request*

Different levels of diagnosability can be activated as required in different phases of a mission. An illustration of the isolation on request concept can be seen in figure 4. There can exist two kinds of ambiguities for local diagnosers, *intra-subsystem ambiguities*: due to unisolable faults in the same subsystem, and *inter-subsystem ambiguities*: due to faults which propogate between subsystems causing residual generators to be triggered

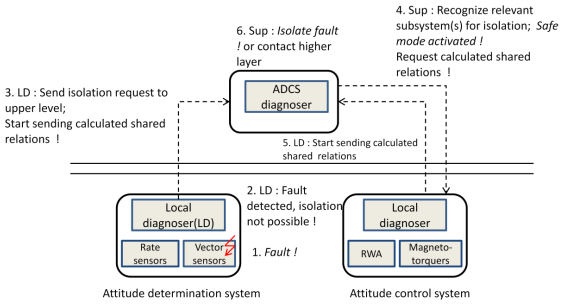

Figure 4. The isolation on request concept

in multiple subsystems. Figure 4 illustrates an example of intra-subsystem ambiguity between faults in two kinds of sensors in the attitude determination subsystem. This illustration is based on the satellite ADCS discussed in section 5, where this ambiguity is shown to exist. The LD_{ADS} raises an isolation request with its supervisor, which can use consistency information from the LD_{ACS} in the form of partially calculated shared relations to disambiguate the fault in the ADS.

We can therefore save on computational resources and communcation bandwidth considerably during nominal operation.

This mode of functioning is critical in the space domain for two reasons. Firstly, faults are rare interruptions of nominal functioning. As such, a much more favourable cost-benefit tradeoff for integrating MBD into a fault protection system can be achieved by keeping computation and communication overheads associated with the diagnoser during nominal operation to the minimum. Secondly, the operation philosophy of space systems and missions is highly conservative by necessity. Fault detection can trigger a switch to safe hold mode, with autonomous isolation and possibly reconfiguration an optional second step. The criticality level of functions and components can be deduced using engineering studies such FMEA and FTA and also the autonomy level required during a certain phase.

3 The Attitude Determination and Control system

The ADCS maintains the desired attitude of the satellite with respect to a reference frame. The ADCS can be functionally decomposed into an attitude determination system (ADS) responsible for estimating the state of the satellite and an attitude control system (ACS) responsible for maintaining the specified attitude utilizing control algorithms and actuators.

The attitude is sensed using rate and vector sensors. State estimation is performed by fusing the sensed quantities. A combination of different actuators is utilized by the ADCS to meet requirements during different mission phases Sidi (1997). In this work we model an ADCS composed of reaction wheels, thrusters and magnetorquers. We have built a simulation testbed in Matlab/Simulink for the configuration of an earth observation satellite in low earth orbit.

For designing the diagnoser, the structure of the ADCS is specified as a set of constraints relating sets of unobserved and observed variables.

Most of the constraints C are composed of three behavioural relations corresponding to the three axes. The decomposition of the structure into the ADS and ACS subsystems is illustrated in figure 5. This structural model is the input for our decentralized architecture and algorithms. While the constraints and variables

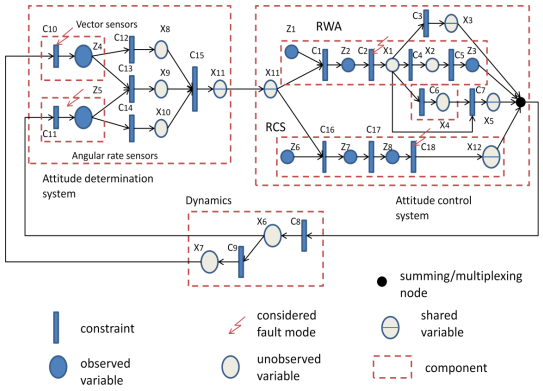

Figure 5. Structural modeling of the ADCS

representing the dynamics of the satellite are shown separately, we consider them part of the ADS in section 4.

In 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 not directly sensed. However estimated states are calculated quantities and can be available for diagnosis. The general procedure for diagnoser design starts with assuming the smallest set of directly sensed quantities, which is expanded to fulfill dignosability and isolability specifications if required.

Unobserved Variable	**Description**
$\dot{h}_w/x_1$	flywheel momentum derivative
h_w/x_3	Flywheel angular momentum
ω_w/x_2	Flywheel angular speed
T_m/x_4	Magnetic torque
$T_{RWAtotal}/x_5$	Total RWA torque
$\mathcal{X}_\omega/x_6$	angular rates
$\mathcal{X}_{pos}/x_7$	attitude angles
$\mathcal{X}_{est1}/x_8$	Estimated state: vector sensors alone
$\mathcal{X}_{est2}/x_9$	Estimated state: rate and vector sensors
$\mathcal{X}_{est3}/x_{10}$	Estimated state with rate sensors
$\mathcal{X}_{est}/x_{11}$	Estimated state
$T_{RCS\,total}/x_{12}$	Net RCS torque

Observed Variable	**Description**
$\mathcal{X}_{RWAref}/z_1$	RWA state reference
T_{RWAc}/z_2	RWA control torques
$\hat{\omega}_w/z_3$	Sensed value of flywheel angular speed
$\hat{\mathcal{X}}_\omega/z_5$	Sensed angular rates
$\hat{\mathcal{X}}_{pos}/z_4$	Sensed attitude angles
$\mathcal{X}_{RCS\,ref}/z_6$	RCS state reference
$T_{RCS\,c}/z_7$	RCS control torques
$T_{RCS\,scaled}/z_8$	distributed and scaled torques

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	$frw(frw_x, frw_y, frw_z)$

While the models as summarized include both RCS and RWA based control, further discussion of diagnoser design is based on the satellite operating in a mission mode when RWA based control is utilized. This submodel of the ADCS is composed of 42 equations in total with 42 unobserved variables, 15 observed variables and 9 additive faults which are modeled as variables in the equations.

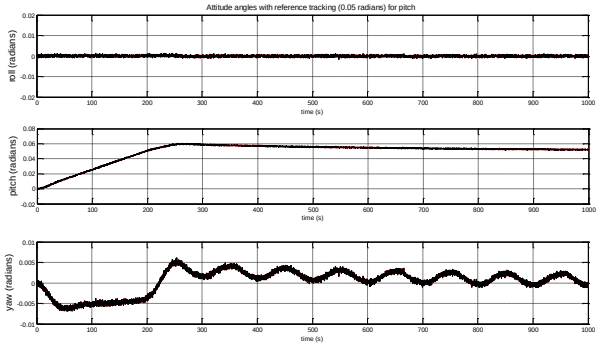

Figure 6. Attitude angles with reference tracking for pitch axis

4 Diagnoser Design and Implementation

This section describes the design and implementation of the decentralized diagnoser for the satellite ADCS. The structural model of the ADCS is used to design the diagnoser. This discussion is summarized from S. Indra (2011). Next, we will select the ARRs to implement and use the algorithm introduced in Svard and Nyberg (2010) to derive the ARRs. Simulation results with fault injection demonstrate an example of isolation on request.

Diagnoser Design

It was shown in S. Indra (2011) that a centralized diagnoser for the ADCS would ensure complete fault isolability for single faults. It was also shown that designing local diagnosers for the ADS and ACS seperately without deriving shared relations leads to a loss of isolability for ADS faults. The isolability could be recovered by applying the proposed decentralized architecture and designing local & supervisory diagnosers. From the standpoint of the local diagnosers, the shared variables X^{shared} are then assumed to be observed.

The ADS local diagnoser without considering X^{shared} observed has a *maximum fault isolability* of $\{frs_x, fvs_x\}, \{frs_y, fvs_y\}, \{frs_z, fvs_z\}$.

We observe the intra-subsystem ambiguity among the rate and vector sensor faults in the ADS.

The ADS local diagnoser considering X^{shared} observed has a *maximum fault isolability* of $\{frs_x\}, \{frs_y\}, \{frs_z\}, \{fvs_x\}, \{fvs_y\}, \{fvs_z\}$. ACS local diagnoser considering X^{shared} observed has a *maximum fault isolability* of $\{frw_x\}, \{frw_y\}, \{frw_z\}$. For the ADCS supervisory diagnoser to disambiguate faults the *interesting fault vector* is $[frw_x, frw_y, frw_z, frs_x, fvs_x]$ and complete isolability is possible with a *maximum fault isolability* of $\{frw_x\}, \{frw_y\}, \{frw_z\}, \{frs_x\}, \{fvs_x\}$. The supervisory diagnoser is able to differentiate between faults frs_x and fvs_x.

Diagnoser Implementation: We derive analytical expressions for the local diagnoser and supervisory diagnoser using the algorithm of Svard and Nyberg (2010). This algorithm derives a *sequential residual generator* from a redundancy which is structurally present.

The local residual generators are implemented in the local diagnoser. The ADS diagnoser will not however be able to isolate between faults frs_x and fvs_x. This can be achieved with the hierarchical residual generators.

These residual generators were implemented in the simulation

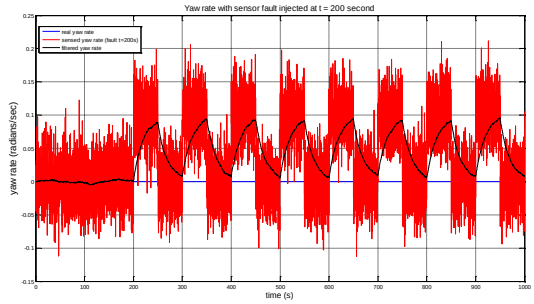

Figure 7. Raw sensed, filtered and real yaw rates with intermittent offset fault injected into sensor at 200 seconds

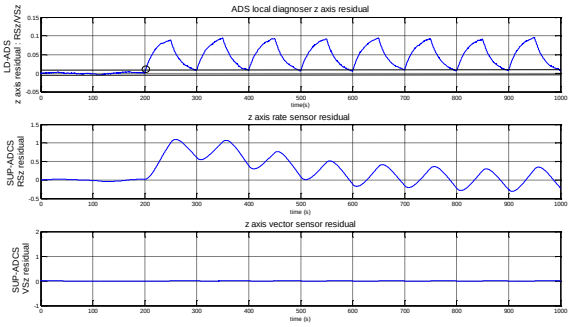

Figure 8. The supervisor is triggered after fault detection by the ADS local diagnoser : Isolation on request

in a mode when the satellite is controlled with reaction wheels. During reference tracking on the pitch axis as seen in figure 6, an intermittent offset fault was injected in the yaw rate sensor at 200 seconds. The isolation on request procedure can be observed in figures 7 and 8. The first subplot in figure 8 shows the output of the ADS local diagnoser, which detects a fault but can not isolate it between the yaw rate and vector sensors. The supervisory diagnoser is triggered, and isolates the fault to the yaw rate sensor.

5 Conclusion

A decentralized diagnosis architecture based on ARRs was developed. The architecture relies on a functional breakdown of the system into subsystems. Such a decomposition provides an elegant way to integrate diagnoser development with subsystem development using a systems engineering framework. Rather than execute a monolithic centralized diagnoser at all time, computation and communciation overheads during nominal operation can be minimized with isolation on request. Further work will focus on how the diagnosability requirements for different mission phases can be derived based on reliability and redundancy studies of components and subsystems. The diagnoser architecture can then be optimized for different mission modes with a given diagnosability specification. Also interesting would be investigating how diagnosers operating with a heterogeneous mix of modelling frameworks for different subsystems could be integrated into the decentralized architecture and work with a common interface.

A simulation testbed for satellite attitude and orbit control system (AOCS) has been developed. This benchmark simulation includes together with the dynamics and control algorithms of an AOCS, component level models of sensors and actuators at varying levels, and also realistic fault scenarios for these components. The rule based fault diagnosis schemes conventionally implemented onboard spacecraft are also included for comparison purposes. After some more integration and testing we plan to release this simulation publicly to the FDI community as a benchmark case study.

References

Armengol, J., Bregon, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., Olive, X., Pulido, B., and Travé-Massuyès, L., "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*, 1480–1485, 2009.

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

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

Krysander, M., Åslund, J., and Nyberg, M., "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*, 38(1), 2008.

Kurien, J. and R-Moreno, M., "Costs and benefits of model-based diagnosis", In *2008 IEEE Aerospace Conference*

Morgan, P.S. (2011). *System Health Management with Aerospace Applications*, Chapter 34, 543–554. John Wiley and Sons, United Kingdom.

Indra, S. , Travé-Massuyès, L. , and Chanthery E., "A decentralized FDI scheme for spacecraft: Bridging the gap between model based FDI research and practice", In *Proceedings of the 4th European Conference for Aerospace Sciences*, Saint Petersburg, Russia, 2011.

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

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

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

Tumer, I.Y., *System Health Management with Aerospace Applications*, Chapter 8, 129–142, John Wiley and Sons, United Kingdom, 2011.