Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on
Guidance, Navigation & Control, Delft University of Technology, Delft,
The Netherlands, April 10-12, 2013

Chapter 1
Spacecraft Fault Detection & Isolation System Design using Decentralized Analytical Redundancy

Saurabh Indra and Louise Travé-Massuyès

Abstract Fault detection and isolation (FDI) functionality constitutes a critical element of spacecraft fault protection system capabilities. The FDI schemes currently implemented on board operational spacecraft suffer from a lack of systematic design methods and resulting behavior. While model based diagnosis techniques can resolve a number of these issues, their applicability to spacecraft has been limited until now largely due to an unfavorable net value proposition. An approach integrating analytical redundancy based diagnosis into a conventional spacecraft FPS architecture is presented. The approach is based on a novel decentralized diagnosis architecture based on analytical redundancy relations. A systematic approach to designing such decentralized model based diagnosers for spacecraft is discussed, with a focus on the attitude and orbit control system. Analytical redundancy relation based error monitors and activation rules relying on the corresponding fault signatures are derived during the design phase. A comparison with the diagnosis functionality as currently implemented in the Cassini attitude and articulation control system fault protection is presented in terms of the design & development effort. It is demonstrated that the presented diagnoser design approach addresses several issues with the conventional methods, while having reasonable additional costs

1.1 Introduction

The space missions of the future envisage autonomous spacecraft operation in challenging environments. Robust and capable fault protection is an enabling technology for such missions. Fault protection is a mix of hardware and software mechanisms

Saurabh Indra
LAAS-CNRS, 7. Avenue du Colonel Roche, 31400 Toulouse, France , e-mail: sindra@laas.fr

Louise Trave-Massuyes
LAAS-CNRS, 7. Avenue du Colonel Roche, 31400 Toulouse, France ,e-mail: louise@laas.fr

aiming to increase the robustness of space systems. The elements of a fault protection system which detect and (possibly) isolate faults constitute the diagnoser.

Traditionally, fault diagnosis onboard spacecraft has relied on rule based techniques. Most of the fault monitors utilized rely on simple mappings from observed symptoms to probable diagnosis, with other techniques being used on a case to case basis. Certain key variables of the system are monitored, and a fault is signalled when the variable is out of the expected nominal range. *Activation rules* respond to subsets of the error monitor outputs and diagnose the cause of anomalous behaviour at the component or functional level. This reliance on *symptoms* instead of an *underlying model of behavior* leads to opacity of structure and behavior. The possibility of different symptoms triggered by the same underlying fault, different priorities among faults, mission modes and other system wide considerations have to be taken into account. Such considerations lead to a *patchwork* of monitors, activation rules and the parameter sets associated with them.

With increasing ambitions for space missions and the associated rise in space system complexity, scaling up such rule based diagnosers is proving difficult. The core issue is the lack of transparency in requirements, design, structure and resulting behavior as discussed by Rasmussen [1].

In contrast to rule based methods, the basic principle of model based diagnosis (MBD) is to use a model of the system with sensed observations from the real system to detect and isolate faults. Basing diagnosis decisions on a system model can address some of the crucial scalability and structural transparency issues associated with rule based diagnosers. It would seem then, that utilizing model based techniques could lead to more effective fault protection systems. However the actual use of MBD techniques has been constrained due to the high associated costs and risks relative to the benefits provided.

There are two main streams of MBD, originating from different communities. While the DX or consistency based approach originates from work in the computer science and artificial intelligence areas, the FDI stream is rooted in systems and control theory. Livingstone and Livingstone 2, flown as experiments onboard the Deep Space 1 and Earth Observor 1 spacecraft are examples of diagnosers based on the DX approach to MBD. The two streams emphasize different diagnosis functionalities. However, there has not been significant mission pull for adoption of such consistency based techniques for fault diagnosis onboard operational spacecraft since then. The unfavourable net value proposition causing this gap between the promise and reality of this stream of MBD is discussed in Kurien & Moreno [2].

Analytical redundancy based MBD on the other hand is a technique utilizing the FDI approach [6]. Using observers to model nominal and faulty system dynamics is one way to realize analytical redundancy. An early theoretical survey of these techniques and their utility for aerospace systems can be found in Patton[7] and there are various operational examples[8].

The second route to implementing analytical redundancy is based on analytical redundancy relations (ARRs). This technique is based on using sensing/structural redundancies in a system to compile consistency checks known as ARRs *offline*. These ARRs are then evaluated online as residual generators using sensed quantities from the system. We utilize as starting point in our work an ARR based approach to diagnosis based on an algorithm discussed in Krysander et al.[9], and extended in Krysander et al[16].

The underlying concepts and approaches of the DX and FDI streams have recently been compared and proved to be equivalent under certain conditions Cordier et al.[10]. However the emphasis in diagnosis functionality and conditions of their optimal usage are different.

One of the most complex and capable FPS operational in space was developed for the Cassini spacecraft and can be considered illustrative of the state of practice of conventional design for interplanetary probes. In this paper we will use this FPS both to illustrate the challenges involved in FPS design, implementation and operation and as a benchmark to assess the value of applying our decentralized diagnosis architecture. The driving system level FP considerations for the Cassini spacecraft are discussed in Slonsky[17].

Our diagnosis approach is based on ARRs and is therefore relevant for continuous state systems modelled for example as a system of differential-algebric equation (DAE) models or as state space models. Most of components of the attitude and orbit control system are usually modeled in such frameworks. Therefore we concentrate in particular on the subsystem level FP operating in the attitude and articulation control system of the Cassini as discussed in Brown et al.[12].

Instead of utilizing a patchwork of different techniques for the design of fault monitors for different faults as discussed in Lee[13] and Macala[**?**], the presented integrated design method utilizes a structural model of the ADCS to derive ARR based fault monitors. The fault signatures associated with these monitors are also derived during the design phase. The approach is based on a novel decentralized ARR based diagnosis architecture. The *hierarchically scalable* nature of the architecture allows systematic design and analysis of fault monitors for different monitoring levels. The architecture addresses some of the structural and behavioral transparency issues as discussed in Rasmussen [1] and Slonski [**?**]. Additionally, the net value proposition of the ARR based diagnosers in terms of benefits and costs is demonstrated to be positive compared to the conventional approaches.

The paper is structured as follows. The issues with conventional FPS design are described in section 1.2, utilizing the fault protection of the attitude and articulation control system of the Cassini as a case study. Section 1.3 starts with an discussion of the ARR based approach to diagnosis followed by a description of the decentralized diagnosis architecture. A comparison between this architecture and the conventional diagnosis techniques used for the Cassini is then provided in section 1.4. The paper concludes with a discussion of the contribution and perspective for future work in section 1.5.

1.2 Fault Protection Systems

Mechanisms and strategies implemented on board spacecraft for increased robustness constitute fault protection. The scope and sophistication of onboard FP functionality is determined by mission specific considerations such as the autonomy level required onboard, communication possibilities with the ground segment etc.

Most spacecraft implement standard FP functions to respond to system level effects. Safe mode responses configure the spacecraft to a power positive, thermally safe state. The safe mode also ensures the availability of a robust link with the ground segment, so that the ground segment has access to housekeeping telemetry. Other examples of standard FP strategies are the command-loss and under-voltage responses. Besides these standard system level functions, subsystem level FP is also implemented depending upon the complexity of the spacecraft and mission requirements. The Cassini FP aims to ensure robustness of the mission to all probable single point failures. We focus on the subsystem level fault protection of the attitude and articulation control system (AACS) in the following discussion.

The conventional *monitor-response architecture* forms the basis of the AACS fault protection system. This structure is illustrated in the figure 1.1 [12]. *Error monitors* and *activation rules* make up the diagnosis elements, while *response scripts* and the *repair manager* implement the reconfiguration functionality. Monitors compare sensed values of quantities to expected values and output a health status. Activation rules use subsets of monitor outputs together with the hardware configuration and activity goals to diagnose the likely fault(s).

It is interesting to study the techniques used to implement fault monitors for the different components and control loops of the AACS as illustrated in figure 1.2. This serves to illustrate the wide range of underlying diagnosis techniques of these fault monitors. Component level monitoring is provided by thresholds on individual quantities such as reaction wheel drag. Monitoring at the control loop level is implemented using the control error and its derivative by monitors known as state-space fault monitors. The functioning of the loop is classified as *acceptable* if the control error is below a specified threshold. If the error is reasonably small and decreasing, the functioning is *tolerable*. Large errors which are not decreasing indicate faulty functioning of the control loop. These monitors are a simple form of model based diagnosis as there is a model encapsulated in the controller trying to minimize the loop error. Such monitors run *piggyback* on the model for control instead of a diag-

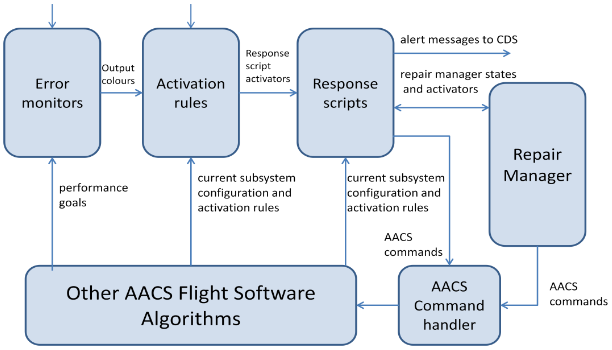

Error monitors
Output colours
Activation rules
Response script activators
Response scripts
alert messages to CDS
repair manager states and activators
Repair Manager
performance goals
current subsystem configuration and activation rules
current subsystem configuration and activation rules
AACS commands
AACS commands
Other AACS Flight Software Algorithms
AACS Command handler

Fig. 1.1 The structure of the Cassini attitude and articulation control subsystem level fault protection

nostic model developed seperately, with its additional costs.

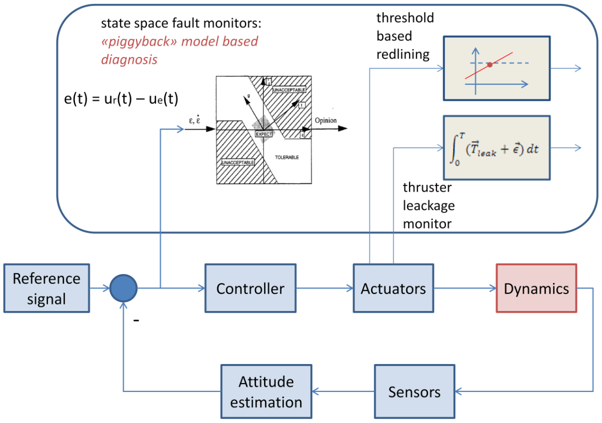

state space fault monitors:
«piggyback» model based diagnosis
e(t) = ur(t) – ue(t)
ε, ε̇
UNACCEPTABLE
EXPECT
TOLERABLE
UNACCEPTABLE
Opinion
threshold based redlining
$\int_0^T (\vec{T}_{leak} + \vec{\epsilon})\,dt$
thruster leackage monitor
Reference signal
-
Controller
Actuators
Dynamics
Attitude estimation
Sensors

Fig. 1.2 Different techniques utilized for the Cassini AACS FPS

The thruster leackage detection monitor is required to detect a leak on one of the eight primary thrusters. Such a leak will cause fuel wastage due to the compensating control which will be triggered. Such leaks need to be detected even while the spacecraft is executing a maneuver. A state space monitor could not be designed as there is no one quantity in the control loop which could signal such a fault during a maneuver. So a model based approach, relying on monitoring deviations from the expected dynamics of the spacecraft was used instead[13]. The resulting thruster leackage monitors are analogous to analytical redundancy relations.

We identify now the issues with the conventional FPS structure and development techniques. The basic problem is the usage of various diagnosis techniques and associated analysis methods depending on the detection requirements on a case to case basis. The techniques utilized for the Cassini AACS fault monitors range from rule to model based methods as discussed earlier and seen in figure 1.2. While the various forms of analysis required for each of the techniques adds to the development effort, the resulting structure of the diagnosis elements suffers from a *lack of architectural pattern*. The lack of an integrated architecture complicates the task of setting parameters and working out activation rules. This effort is shifted to a large extent to an ad hoc one based on simulation. With increasing system complexity such an approach does not scale well, leading to opacity of diagnoser structure and behaviour, the possibilty of *emergent behavior*, and consequently lower robustness.

These issues were detailed in Rasmussen[1] and Slonsky[17] and are summarized briefly below.

- Absence of architectural pattern. A bottoms up approach of fitting techniques to requirements & problems on a case to case basis.
- Lack of explicit models explaining what caused a monitor to be triggered.
- Distribution of state and behavioral information among complex parameter sets - thresholds, persistence counters, disable/enable flags, timers etc.

Many of these issues are connected to the special situation of fault protection functionality as compared to other functional subsystems like AOCS and propulsion. Health management functionality for a system is a *set of capabilities* spread over the functional subsystems. However it is also necessary to view FP capability as constituting a *virtual subsystem* in its own right, because the interactions among capabilities built into seperate subsystems should be worked out as early as possible, and sound systems engineering practices followed during development and testing.

The decentralized architecture developed in this thesis can serve to address a few of these challenges, and this attempt is described in the following sections with the decentralized architecture itself and then with its application to the Cassini FPS.

1.3 Decentralized diagnosis with analytical redundancy relations

In order to describe the decentralized diagnosis architecture based on ARRs, we introduce first the basic notions associated with the structural approach to ARRs. Given the emphasis of this paper, and the space constraints, both this introduction and the following description of the decentralized architecture are developed in an *intutive* rather than formal fashion. The reader can look to the references for formal description of the concepts involved.

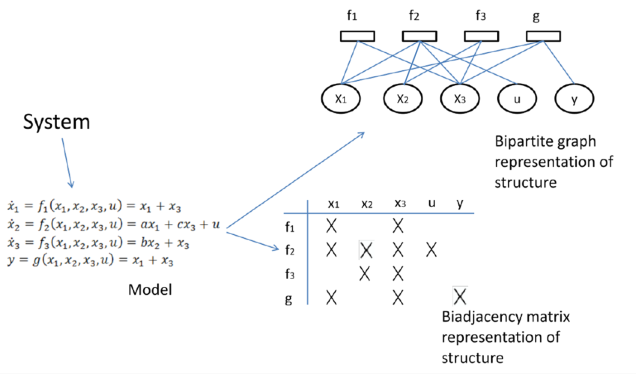
1.3.1 The structural approach to analytical redundancy relations

Analytical redundancy relations rely on using redundancies in the system to compile consistency checks known as residual generators offline. The particular approach to ARR based diagnosis utilized here is based on designing **residual generators** based on **structural** redundancies in the system. These residual generators serve as consistency checks, using sensed quantites from the system to check whether monitored sections of the system are functioning nominally. A residual generator takes as input the values of the observed variables and, in an ideal case i.e without noise and disturbances, gives a non-zero output when the system behaviour is inconsistent with the model. A detailed description of the structural approach to ARRs can be found in .

The process of deriving ARRs begins with a model of the system in the form of a system description as seen in figure 1.3. The system description consists of a set of equations involving a set of variables. The set of variables is partitioned into a set of *known (or observed)* variables denoted as $\mathbf{Z}$ and a set *unknown (or unobserved)* variables denoted as $\mathbf{X}$. The number of observed variables is $\mathbf{n_Z}$ while the number of unobserved variables is $\mathbf{n_X}$. We refer in the following discussion to the vector of known variables as $\mathbf{z}$ and the vector of unknown variables as $\mathbf{x}$. The system description or model, denoted as $\mathbf{M}(\mathbf{z},\mathbf{x})$ or $\mathbf{M}$, is then a set of equations relating the known variables $\mathbf{z}$ and the unknown variables $\mathbf{x}$. The class of models considered here are *differential-algebric equation* (DAE) systems. Therefore the equations $m_i(z,x) \subseteq M(z,x)$, $i = 1,\ldots,n$, are differential or algebraic equations in $\mathbf{z}$ and $\mathbf{x}$. For the model in the figure 1.3 $\{x_1,x_2,x_3\}$ is the set of unobserved variables, while $\{u,y\}$ is the set of observed variables. Obtaining ARRs for a model $M(z,x)$ involves the elimination of unobserved variables to arrive at a consistency check which can be evaluated based on the sensed quantities.

The *structure* of a system is a representation of which variables are involved in the equations which make up the model of the system. Such a structural abstraction allows us to derive redundancies disregarding the actual analytical expressions of the equations making up the system model. Ignoring the analytical expressions enables the consideration of nonlinear systems, and the use of efficient algorithms while deriving *possible* redundancies. However, the results obtained with such a structural representation are best case scenarios. Causality considerations and the algebraic and differential loops in the DAE system determine which of the theoretically possible structural redundancies can in fact be exploited for the derivation of residual generators. A variable elimination technique and procedure must then be utilized to derive a residual generator involving only observed variables.

A *bipartite graph* can be used to represent the structure of the system and deduce possible paths for variable substitution. To define a bipartite graph representation of the structure of a system let us denote the sets of vertices as $\mathscr{C}$ and $\mathscr{V}$, representing the set of constraints and the set of variables respectively. A vertex $c_i \in \mathscr{C}$ is con-


f1
f2
f3
g
X1
X2
X3
u
y
System
Bipartite graph representation of structure
Model
Biadjacency matrix representation of structure

$\dot{x}_1 = f_1(x_1,x_2,x_3,u) = x_1 + x_3$
$\dot{x}_2 = f_2(x_1,x_2,x_3,u) = ax_1 + cx_3 + u$
$\dot{x}_3 = f_3(x_1,x_2,x_3,u) = bx_2 + x_3$
$y = g(x_1,x_2,x_3,u) = x_1 + x_3$

	x1	x2	x3	u	y
f1	X		X		
f2	X	X	X	X	
f3		X	X		
g	X		X		X

Fig. 1.3 Structural Modeling of a System

nected by an edge to the vertex $v_j \in \mathscr{V}$ if and only if the constraint c_i involves the variable v_j. Referring to the system model $M(z,x)$ introduced above, the equations $m_i(z,x) \subseteq M(z,x)$, $i = 1,\ldots,n$ constitute the set of constraints (C). The set of variables $\mathscr{V}$ is composed of the sensed and unsensed variables $\mathscr{V} = \mathscr{Z} \cup \mathscr{X}$. However for the purpose of finding substitution paths, it is interesting to consider the bipartite graph between the model equations and the unobserved variables - i.e. $\mathscr{V} = \mathscr{X}$.

It can be shown that ARRs correspond structurally to so called *complete matchings* between X and $\mathscr{C}$ on the bipartite graph $G(M \cup X \cup Z, \mathscr{A})$, or equivalently on $G(M \cup X, A)$, where $A \subseteq \mathscr{A}$ and A is a set of arcs such that $a(i,j) \in A$ if and only if variable x_i is involved in relation m_j [6]. A complete matching between X and M, provides a structural path to eliminate the unobserved variables and arrive at a consistency check. A complete matching is denoted as $\mathscr{M}(X,M)$, or simply $\mathscr{M}$ in case there is no ambiguity.

Equivalently, ARRs correspond to **minimal structurally over determined** (MSO) sets, which are sets of equations of the system with one more equation than unknowns [9]. Unobserved variables can be eliminated, and then the redundant equation can be used to check for consistency as seen in figure 1.4. While complete matchings on bipartite graphs provide an intutive, graphical view of structural redundancies, the biadjacency matrix and MSO sets approach is used to implement efficient algorithms.

The number of MSO sets increases exponentially with the degree of structural redundancy present in the system. Rather than deriving all possible MSO sets, the idea of minimal test equation support (MTES), was introduced in [] to limit the derived structural redundancies to those responsive to a set of interesting faults to be considered. Corresponding to each MTES the corresponding fault sensitivity can also be derived using the algorithm presented in [].

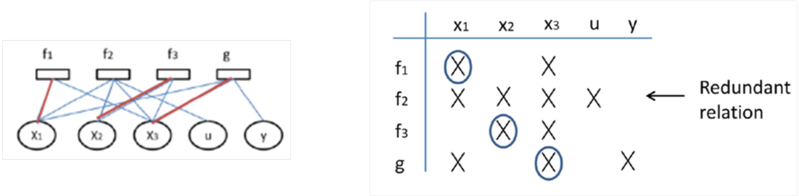


Fig. 1.4 The presence of redundancy in a structural sense: A Minimal Structurally Overdetermined (MSO) Set and a Complete Matching

The (centralized) diagnosis scheme based on analytical redundancy relations can be seen in figure 1.5. The structural model of the system serves as input to the diagnoser design phase. An MSO or minimal test equation support (MTES) signifies the theoretical presence of a structural redundancy which could be used to develop a consistency check for a part of the system. The corresponding minimal test support (MTS) represents the faults which can be detected with this consistency check. In this way the MTS sets characterize the maximum possible fault isolability. Whether a residual generator can be analytically derived depends upon the causality restrictions on the equations in the set and the presence of algebraic and differential loops. We use in our work the residual generator derivation method proposed in [14]. This method relies on deriving 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**. After offline design, the diagnoser is implemented as a residual generator bank.

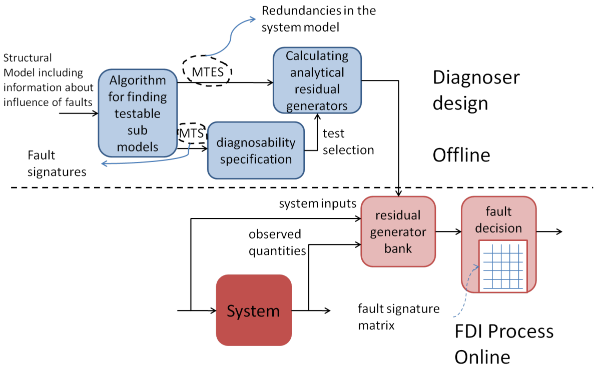

Redundancies in the system model
Structural Model including information about influence of faults
Algorithm for finding testable sub models
MTES
Calculating analytical residual generators
Diagnoser design
MTS
diagnosability specification
test selection
Fault signatures
Offline
system inputs
observed quantities
residual generator bank
fault decision
System
fault signature matrix
FDI Process Online

Fig. 1.5 Diagnosis with Analytical Redundancy Relations

1.3.2 The ARR based decentralized diagnosis architecture

Having discussed briefly the basic notions of the structural approach to ARR based diagnosis, we present intutively the decentralized diagnosis architecture. In this architecture, local diagnosers rely on models of their subsystems to arrive at local diagnosis. Ambiguities might arise as faults propogate between subsystems. A supervisor at the higher level serves to resolve ambiguities and provide diagnosis at a higher resolution than that possible with purely local information. The architecture is hierarchically scalable as can be seen in figure 1.6. This means that the supervisor of one level can act as the local diagnoser for the next higher level.

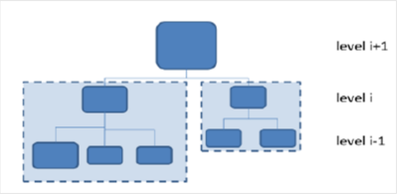


Fig. 1.6 Illustration of the basic diagnoser structure

As discussed earlier, the structural approach to deriving analytical redundancy relations can be viewed as one of finding complete matchings on the bipartite graph representation of the structure. The framework for decentralization relies also on this bipartite graph representation.

The following model is used to illustrate the notions. It is composed of six equations r_{1-6} relating the unobserved variables $X = \{x_1, x_2, x_3, x_4, x_5\}$ and the observed variables $Z = \{u, v, w\}$.

$$r_1 : \dot{x}_1 = -x_1^2 + x_3 + u \tag{1.1}$$

$$r_2 : \dot{x}_2 = x_4^2 \tag{1.2}$$

$$r_3 : x_1 = 3 \cdot x_2^3 + v \tag{1.3}$$

$$r_4 : y = x_4 + x_5 \tag{1.4}$$

$$r_5 : \ddot{x}_3 = x_4^2 + x_5 \tag{1.5}$$

$$r_6 : w = x_5 \tag{1.6}$$

The biadjacency matrix and bipartite graph representations of the structure of this model can be seen in figure 1.8.

To introduce the notions behind the decentralized architecture intutively, consider the system is divided into two subsystems as in figure 1.7. While the variables x_1, x_2 are *local* to subsystem 1 and the variable x_5 local to subsystem 2, the variables x_3, x_4 are *shared* between the two subsystems. The set of variables is therefore divided into local and shared variables.

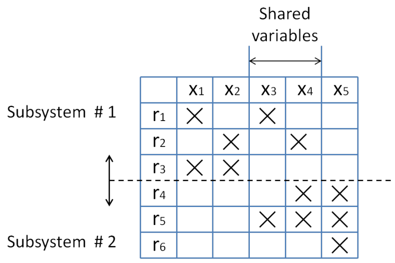


Fig. 1.7 Divison of a system model into subsystems

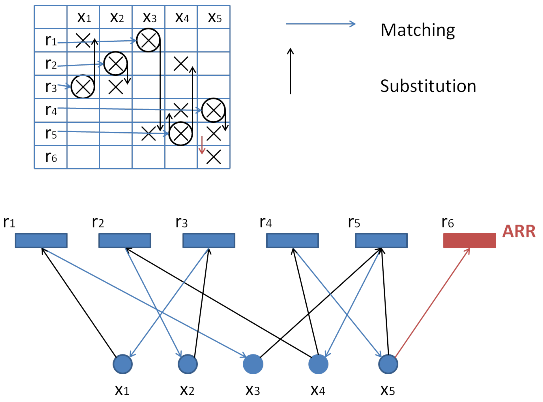


Fig. 1.8 Structural derivation of a redundancy relation at the global level

A complete matching for the global system can be seen in figure 1.8 both on the bipartite graph and the biadjacency matrix. The use of the complete to eliminate all unknown variables is also illustrated with a series of matchings and substitutions. The relation r_6 is used as the redundant relation to serve as the consistency check. Also observe that the sensed variables u, v, y, w are only considered implicitly in the structural representations.

Now consider the situation when we try to use the structural representation of the subsystems as available to the local diagnosers working on the two subsystems as seen in figure 1.8. The concepts of local complete matchings and shared relations have been formalized in [3] and [4]. From the perspective of a local diagnoser, while local complete matchings involve only unknown variables local to subsystems and sensed variables, relations involving shared variables can not be evaluated at that level. Such so called *hierarchical* relations are sent to the supervisory level, where other subsystems also send their hierarchical relations. The supervisory layer attempts to eliminate the unknown variables at its level using these hierarchical relations and arrive at a consistency check if possible. It has been shown in [5], that such a decentralized diagnoser is equivalent from the point of diagnosability to

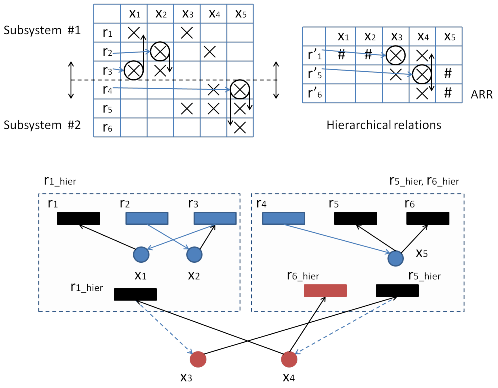


Fig. 1.9 Structural derivation of decentralized local and hierarchical redundancy relations

a centralized diagnoser even if the choice of local matchings differ from the ones used in the global case. This means that the same set of ARRs will be available in both the decentralized and centralized diagnosers from a structural perspective.

1.4 Application to the Cassini Attitude Control System: A Qualitative Comparison

We present in this section a qualitative comparison between the conventional and ARR based diagnosers in terms of the design and development effort. The application of decentralized ARR based diagnosis to the Cassini AACS is used to facilitate this comparison.

The community developing a class of methods and techniques tends to take a relatively narrow view encompassing only the quantitative technical benefits of the methods. However, the decision of whether to use a novel technique for an actual spacecraft and mission is determined by a much broader costs, benefit and risk analysis. It is these *net value* considerations which often serve as bottlenecks in the adoption of new techniques such as MBD.

The challenge of comparing design and development methods in terms of their net value arises from the subjective nature of the considerations involved. However some traction can be obtained by structuring the discussion around the key factors which influence the effort involved in *diagnoser design* and *verification and validation* $(V\&V)$. Therefore the following discussion will be structured around the following two factors :

- The models used for diagnoser design
- The diagnoser design process

1.4.1 Models used for diagnoser design

Attempting to *unravel* the influences and factors involved in the design procedure, we proceed by first discussing the *inputs* to the process - the *models used for diagnoser design*.

A model is typically a set of instructions, controls, equations or constraints which encapsulate knowledge about the expected behavior of a system. Models are abstractions of reality, with a limited range of validity. Expected behavior is always modeled at a certain level of granuality, and in a certain framework.

However in a more general sense, any knowledge about the expected behavior of a system can be considered an implicit *model* of the system. A diagnosis results from reasoning about the expected behavior of a system in the form of a model. However as a model is always an approximate description of the behavior of a system, it has to be made to fit and then validated with real data. The tunable parameters allow the model to be adjusted to fit data from the actual system. A critical distinction therefore needs to be made between the *model structure* and the *model parameters*. To account for the unmodeled dynamics, i.e. behavior not accounted for by the model, *thresholds* are used. This is the case for example with noise and disturbances.

So the distinction between model based diagnosis techniques and conventional rule based methods does not lie in the *presence* of a model, but rather in the utilization of *explicit* models with sophisticated structures. Traditional rule based diagnosis techniques such as thresholding and state-space monitors use very simple model structures, and then rely on model parameters and thresholds to achieve satisfactory response to actual behaviour.

Modeling is always performed for a certain *purpose*, which dictates the aspects which need to be modeled, and also the required granuality. Different models are required for example for simulation and for controller design. Given the considerable effort involved in modeling, keeping modeling costs down is a driving factor when considering the use of new techniques. The conventional error monitor based approach uses the *expected* behavior of the error signal of the control loop encapsulated in the 'state-space' representation of figure 1.10 as a simple model. The *qualitative* status - expected/unacceptable/tolerable of the control loop is determined based on the behavior of the error signal and its derivative. A fault on any of the components in the control loop can affect the error signal, and consequently the monitors.

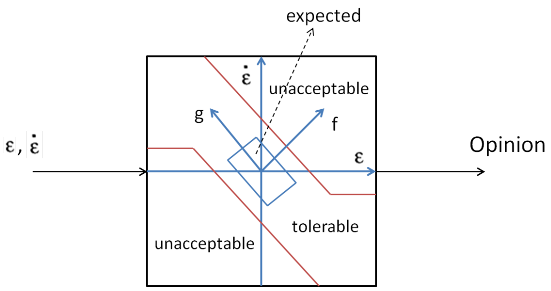


Fig. 1.10 Regions on the 'State Space' plane model the behaviour of the error signal. In effect modeling effort to the setting of the parameters

What about the models used for diagnoser design with the decentralized ARR based approach presented in this thesis ? The structural model utilized contains information about the constraints and variables involved in the system. An example of such a structural model can be seen in figure 1.11 Information about sensor placement is take into account by making the distinction between observed and unobserved variables. While these models are more sophisticated than the simple 'state-space' models, the information encapsulated in them is *conceptually* the same as that represented by control and simulation models of the AOCS as can be seen in the constraints and variables in table 1.1. While control and simulation models include the actual analytical expressions of the constraints, structural models represent the same information at a more abstract level. It is possible in principle to extract the structural information from control and simulation models - which are created during the normal engineering procedure.

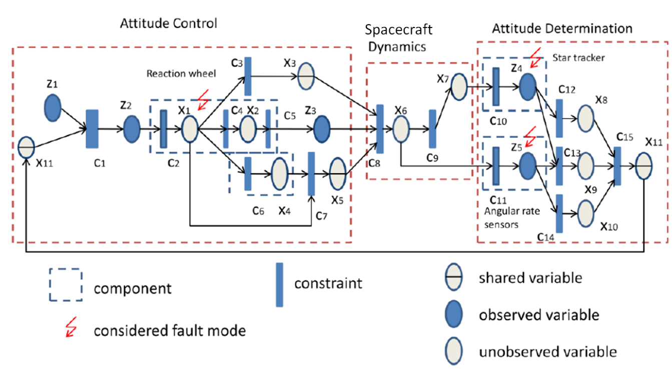


Fig. 1.11 An example of the structural model used for deriving analytical redundancy relations - the model *structure* plays a much greater role

Table 1.1 Example of constraints and variables of the structural model

Constraints	Subsystem	Description
$C_{control}/C_1$	ACS	Control algorithm
C_{RW1}/C_2	ACS	Reaction wheel motor dynamics
C_{RW2}/C_4	ACS	Reaction wheel flywheel dynamics
C_{RW3}/C_3	ACS	Reaction wheel angular momentum integration
C_{dyn}/C_8	DYN (ADS)	Satellite dynamic equations of motion
C_{kin}/C_9	DYN (ADS)	Satellite kinematic equations of motion
C_{RS}/C_{11}	ADS	Rate sensors
C_{VS}/C_{10}	ADS	Vector sensors
C_{est1}/C_{12}	ADS	State estimation with vector sensor alone
$\dot{h}_w/x_1$	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
$\mathscr{X}_{ref}/z_1$	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

In conclusion, how do the two approaches compare then in terms of the model used for diagnoser design ? The model *structure* in the case of the ARR based approach is more sophisticated, but contains the same information as control and simulation models. In the case of the conventional error monitor based approach, the model *structure* is very simple, being defined as regions on a plane. Much more of the behavioral information is defined rather in the model parameters - for example the parameters delimiting the regions considered ’normal’, ’tolerable’ or ’unacceptable’. This is a good example of ”behavioral information being spread over parameter sets” as described by Rasmussen [1].

1.4.2 The diagnoser design process

Having discussed the *input* to the diagnoser design process in the two approaches - i.e. the models utilized for diagnoser design, we consider the design procedure itself. What constitutes diagnoser design ? We define the design process here as the *derivation of the structure* of the monitors which constitute the diagnoser and then the *setting of the diagnoser parameters* to achieve *optimal* diagnosis. An optimal diagnosis for a given diagnoser would achieve the best possible performance in terms of the considered *quantitative metrics*. These quantitative metrics would include detection time performance, false alarm rate and missed detection rates.

A simulation of the system, with realistic noise and disturbance models is used to tune the diagnoser, with the injection of realistic faults. The faults which are considered would result from engineering analysis such as FMEA and FTA.

How do the two approaches compare ? We contrast first the derivation of the structure of the diagnoser, and then the setting of the parameters.

In the case of the ARR based diagnoser, the structural model is utilized as input to an algorithm which identifies the monitorable structural redundancies present in the system, with the possiblity of focusing on a set of interesting faults. Then an automatic derivation of the analytical expressions of the residual generators is possible utilizing for example the algorithm proposed in Svard & Nyberg [14].

In contrast, as the structure of the conventional error monitors is the same for the different components in the loop and various faults, the diagnoser design effort for these monitors consists largely not in the derivation of monitor structure but in parameter tuning which is discussed below. The thruster leackage monitors of the Cassini AACS fault protection are based on ARRs. But they were used only because conventional error monitors were not able to satisfy requirements and their derivation was not an automated process.

In the structural derivation phase therefore, the possibility of *systematic, integrated* design with the decentralized ARR based method provides a significant improvement over the conventional design approach as utilized for the Cassini AACS FPS which consists of a *patchwork* of techniques.

And how about diagnoser parameter settings ? The setting of diagnoser parameters aims to optimize (and trade off between) FDI performance and robustness for a given diagnoser structure. Thresholds, counters and flags are examples of diagnostic system parameters. The effort involved in tuning the diagnostic system is strongly related to the clarity of the physical relation between the parameters to tune and the underlying properties of the system.

The first difference is in terms of the degree and nature of the role of diagnoser parameters. The extent of the role of diagnoser parameters is *inversly* proportional to the sophistication of the model structure utilized for diagnoser design. Due to the very simple model structure utilized in the conventional design approach, fitting the diagnoser behavior to data from the system relies to a large degree on the model parameters. The use of an explicit and relatively sophisticated model in ARR based approaches implies less reliance on parameters.

The second contrast is caused by the fact that the fault sensitivities of the residual generators in the ARR based diagnoser are structurally decoupled and computed in the design phase. The different techniques utilized in the conventional approach could lead to a fault propogating and triggering monitors at different levels and locations. Studies such as FMEA provide a guide to work out the activation rules in this case, followed by simulation runs with fault injection. However, in the decentralized

ARR based approach considerable design effort is shifted from the simulation phase to the design phase with the activation rules automatically derivable from the fault sensitivities of the ARR based fault monitors. We can conclude that the presented approach leads to diagnosers which are much more *transparent* and therefore easier to tune compared to the conventional methods.

1.5 Conclusion

The conventional techniques used to design the diagnosis elements of spacecraft fault protection systems suffer from various issues, severely restricting the scalability of such methods as space systems increase in complexity. These issues are illustrated using the example of the fault protection functionality of the Cassini attitude and articulation control subsystem. We then present a decentralized analytical redundancy relation based diagnosis architecture which can address some of them. The application of this architecture to the Cassini attitude control system is contrasted to the diagnosis elements of the conventional Cassini FPS. The comparison is in terms of qualitative metrics such as diagnoser design effort and system structure. Discussing such qualitative factors is essential as it is ultimately these issues which have restricted the application of model based diagnosis techniques for space systems previously. The benefits of the proposed approach are demonstrated. The decentralized diagnoser enables the deployment of varying levels of diagnosability, which is not possible with a monolithic ARR based diagnoser. In future work we are focusing on possibilities related to optimizing the decentralized diagnoser structure and splitting such decentralized ARR based diagnosers between the space and ground segments.

Acknowledgements The work discussed in this paper is funded by Thales Alenia Space, Cannes and the Centre National d'Études Spatiales. We wish to thank Xavier Olive of TAS and Raymond Sougmagne of CNES for their advice.

References

1. R. Rasmussen, *GNC Fault Protection Fundamentals*. In Proceeding of the American Astronautical Society 31st Annual AAS Guidance and Control Conference, Breckenridge, Colorado, USA, 2008
2. J. Kurien and M.D. R-Moreno, *Costs and Benefits of Model-based Diagnosis*, In Aerospace Conference, 2008 IEEE, March 2008
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 Proceedings of the 4th European Conference for Aerospace Sciences (EUCASS), St. Petersburg, Russia, 2011

4. S. Indra, L. Trave-Massuyes and E. Chanthery, *Decentralized Diagnosis with Isolation on Request for Spacecraft*, In Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess), Mexico City, 2012
5. E. Chanthery, S. Indra and L. Trave-Massuyes *The equivalence of global and decentralised ARRs computation*, LAAS-CNRS Technical Report N11094, March' 2011
6. M. Blanke, M. Kinnaert and J. Lunze. Diagnosis and fault-tolerant control. Springer, 2006.
7. R.J. Patton. *Fault detection and diagnosis in aerospace systems using analytical redundancy*. Computing Control Engineering Journal, vol. 2, no. 3, pages 127 –136, May 1991.
8. Douglas J. Zimpfer. System health management with aerospace applications, chapitre Flight Control Health Management, pages 483–495. John Wiley and Sons, United Kingdom, 2011.
9. M. Krysander, J. Åslund and M. Nyberg. *An Efficient Algorithm for Finding Minimal Overconstrained Sub-systems for Model-based Diagnosis*. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 38(1), 2008.
10. M.-O. Cordier, P. Dague, F. Levy, J. Montmain, M. Staroswiecki and L. Trave-Massuyes. *Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 34, no. 5, pages 2163 –2177, oct. 2004.
11. J. P. Slonski. *System Fault Protection Design for the Cassini Spacecraft*. Rapport technique UW-CS-TR-1481, Jet Propulsion Laboratory, California Institute of Technology, October 1995.
12. G.M. Brown, D.E. Bernard and R.D. Rasmussen. *Attitude and articulation control for the Cassini spacecraft: a fault tolerance overview*. In Digital Avionics Systems Conference, 1995., 14th DASC, pages 184 –192, nov 1995.
13. A. Y. Lee. *A Model-based Thruster Leakage Monitor for the Cassini Spacecraft*. Journal of Guidance, Control and Dynamics, 1998.
14. 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), pages 1310–1328, 2010.
15. H. Praehofer B. P. Zeigler and T. G. Kim. Theory of modeling and simulation. Academic Press, San Diego, California, USA, 2000.
16. M. Krysander, J. Åslund and E. Frisk. *A Structural Algorithm for Finding Testable Submodels and Multiple Fault Isolability Analysis*. 21st International Workshop on Principles of Diagnosis (DX-10), 2010.
17. J. P. Slonski. *System Fault Protection Design for the Cassini Spacecraft*. Rapport technique UW-CS-TR-1481, Jet Propulsion Laboratory, California Institute of Technology, October 1995.