INNOVATIVE FAULT DETECTION, ISOLATION AND RECOVERY STRATEGIES ON-BOARD SPACECRAFT: STATE OF THE ART AND RESEARCH CHALLENGES.

A. Wander & R. Förstner
Bundeswehr University Munich, Institute of Space Technology and Space Applications (LRT9.1), Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

Abstract

This paper summarizes basic concepts and the current state of the art in spacecraft fault detection, isolation and recovery. A gap analysis focuses on drawbacks of classical fault diagnosis methods in deep space applications. Studies that propose innovative techniques are summarized and evaluated briefly with respect to enhanced spacecraft on-board fault diagnosis on system level. Research challenges for the introduction of these methods in spacecraft FDIR are outlined and a development roadmap is elaborated.

1. INTRODUCTION

Fault detection and diagnosis is an important problem in spacecraft operations and a critical aspect of on-board software with respect to safety, performance and reliability. As future space missions include ambitious goals resulting in demanding performance accuracy and availability requirements, spacecraft design becomes highly sophisticated and complex. Ground operations teams must observe increasingly large volumes of telemetry data to diagnose faults coping with possibly incomplete or erroneous data sets. Interplanetary spacecraft missions that experience large Earth-spacecraft distances present an additional challenge, as the increasing time delay between commands sent and received by the spacecraft limits the ability to respond to faults in a timely manner.
Today's rapid progress in computing power enables the transfer of fault detection, isolation and recovery (FDIR) functionality from the ground to the spacecraft itself to ensure safe and high-performance operation with less intervention by human operators. Increasing spacecraft autonomy offers also the prospect of reducing overall operational cost.
Thus, the purpose of the presented research is to enhance spacecraft on-board autonomy by developing advanced fault detection, isolation and recovery techniques. Research and development activities focus on system level FDIR of unmanned spacecraft: implementing innovative mechanisms combined with well-proven methods of conventional FDIR can reduce the number of safe mode events and increase operational time of a spacecraft. Particularly on highly complex interplanetary space missions, intelligent on-board FDIR is crucial in case of unexpected failure.

The paper is organized as follows: First, the terminology used throughout the paper is defined briefly. Today's state of the art of conventional FDIR methods on-board spacecraft is summarized. An evaluation of currently used FDIR methods is performed and improvements are suggested. A study of various techniques in order to enhance spacecraft autonomy is presented and the potential of the various concepts is discussed. Finally, research challenges for future work on innovative FDIR methods are elaborated and a development roadmap is presented.

2. TERMINOLOGY

2.1. Fault Detection, Isolation and Recovery

Since the terms fault and failure are used throughout literature with many different meanings, a definition for the use within this paper is deemed necessary:
A fault is defined as an undesired deviation of at least one characteristic property of a system variable from an acceptable/nominal behavior that leads to degraded overall system performance, malfunctions or failure of the system. A failure denotes the total cessation of a function, via a subsystem or the total system.
There are also many definitions of a fault detection, isolation and recovery (FDIR) mechanism to be found in the literature. Generally, the following tasks are part of a modern FDIR system:

– Fault detection is the determination of the presence of faults in a system and of their times of occurrence. Generally, fault detection is followed by
– Fault isolation to determine the type and location of faults. Subsequently,
– Fault identification aims at determining the size and time-varying behavior of the faults as well as estimating the severity of the fault and its possible effects on the system.
– System reconfiguration compensates the identified faults, e.g. by switching to redundant systems.

Terms also commonly used in the control literature are Fault Detection and Diagnosis (FDD), which includes the tasks of FDIR minus the system reconfiguration or recovery. Fault Detection and Isolation (FDI), however, refers to a health management system that is dedicated to fault detection only without characterizing the observed fault or the resulting effect on the system. The terminology is summarized in FIGURE 1. These definitions are based on [1-4].

2.2. Autonomy and Automation

Autonomy and automation have a wide range of definitions, it seems appropriate to establish how those terms will be used in the context of this paper following

[5]: Both terms refer to processes that will be executed independently without any human intervention. Automated processes follow a step-by-step sequence that may still include human participation. Autonomous processes emulate human processes rather than simply replacing them.

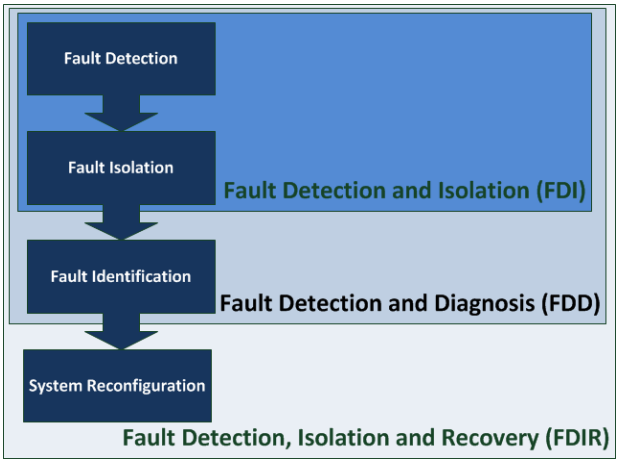

FIGURE 1. Definition of fault detection and isolation (FDI), fault detection and diagnosis (FDD) and fault detection, isolation and recovery (FDIR).

The European Cooperation for Space Standardization (ECSS, [6]) offers a classification of autonomy levels as cited in TAB 1. Following Olive [7], spacecraft operations today reach the autonomy level E2, while the research introduced within this paper aims at autonomy level E4 to be applied to system level FDIR in mission critical situations where ground intervention is not possible, e.g. due to signal delay times or visibility issues.

TAB 1. Mission execution autonomy levels [6].

Level	Description	Functions
E1	Mission execution under ground control; limited on-board capability for safety issues	Real-time control from ground for nominal operations Execution of time-tagged commands for safety issues
E2	Execution of pre-planned, ground-defined, mission operations on-board	Capability to store time-based commands in an on-board scheduler
E3	Execution of adaptive mission operations on-board	Event-based autonomous operations Execution of on-board operations control procedures
E4	Execution of goal-oriented mission operations on-board	Goal-oriented mission re-planning

3. BASIC CONCEPTS IN SPACECRAFT FDIR

Today, a satellite is a smart embedded system that is able to react to some known events and to select a decision among a predefined set [8], [7]. Traditional spacecraft fault management implements redundant hardware and software, functional/analytical redundancy and fault protection algorithms associated with simple consistency checks, voting mechanisms or simple estimation techniques such as Kalman filters. Fixed thresholds are used for rapid recognition of out-of-tolerance condition [2]. These actions of fault detection and isolation on-board are implemented after a careful assessment of possible faults and failure scenarios during design time. The current practice and state of the art to do so are fault tree analysis (FTA) and failure mode, effects and criticality analysis (FMECA) [9]. The resulting FDIR concepts and implementations are described in the following sections.

3.1. Hardware Redundancy

The hardware redundancy concept uses multiple sensors, actuators, processors and software to measure or control a particular value. Typically, fault detection is accomplished by cross checks between redundant units, voting mechanisms and/or component built-in health test techniques like rule-based limit checking. In fault case, possible measures are retry command or reboot, and enabling switching routines on the appropriate FDIR level to switch to either a redundant component (if the faulty equipment was identified) or to a completely redundant string, if no fault identification is performed. In [7], Thales Alenia even reports a “half satellite” strategy, where only fault detection is performed on-board. If a fault is detected, each unit is switched to a redundant one and satellite mode is changed to a predefined system safe mode, waiting for ground intervention.

In order to fulfill failure tolerance requirements, double redundancy is used for essential components such as data management units or reaction control hardware. Single redundancy is sufficient for components that are not safety critical [10]. Cross-strapping redundant hardware is used sometimes in deep space mission like New Horizons [11], to make the spacecraft fully tolerant to almost every single point failure, but requires thorough fault identification to determine a faulty unit, flag it accordingly and avoid its further use.
Dissimilar hardware is used to avoid loss of redundant units due to common failure scenarios.

3.2. Safe Mode

The operational concept of a typical spacecraft includes one or more safe mode configurations that represent the ultimate reaction of the system level FDIR to spacecraft anomalies.
Safe mode (or survival mode) is characterized by configuration of a spacecraft in which it can remain safely without ground segment intervention for a specified period of time. In safe mode, typically all non-essential on-board units or subsystems are powered off, either to conserve power or to avoid interference with other subsystems, and the spacecraft is (automatically) oriented to a particular attitude (thermally safe) with respect to the Earth or the sun. [12] Either particular or all subsystems are switched to redundant hardware strings. It is crucial for the spacecraft to establish link to the ground station crew, since the recovery of the spacecraft from safe mode to nominal mode needs to be commanded by ground [13].
In mission critical phases, where transition to safe mode would lead to degraded science return or mission loss, the safe mode is switched off.

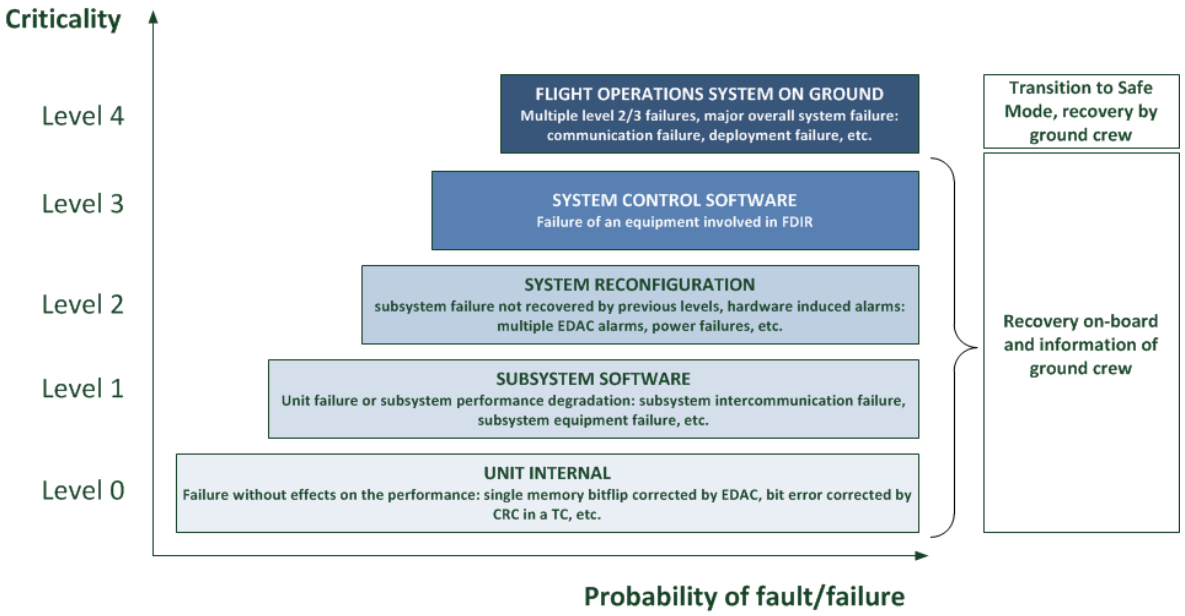

FIGURE 2. Hierarchical FDIR concept following [7], [14] and [13].

3.3. Hierarchical FDIR strategies

For technical, development and programmatic reasons, traditional FDIR functions of a spacecraft are arranged in a hierarchical architecture as depicted in FIGURE 2. Several levels of faults are defined from local component/equipment/unit level up to global system failures. The higher the level, the more critical the fault but lower the occurrence probability of the fault.

Fault detection is implemented on unit level (level 0), fault isolation, if any, is implemented on subsystem or system level (level 2 or 3) and fault recovery is implemented on either relevant level as low as possible. Fault recovery or system reconfiguration means, switching to (hot) redundant units/strings or change operational mode to safe mode. Redundancy is managed in an upper layer to provide a comprehensible decision and track unit faults and failures that already occurred.
TAB 2 shows a summary of fault impacts on the system according to the level the fault occurs, how the fault is detected and which recovery actions can be taken by the spacecraft control system.
The implementation of recovery actions in modern spacecraft of the European Space Agency (ESA) is based on preprogrammed On-Board Control Procedures (OBCPs) that represent the system's event-triggered reflex reaction to FDIR alarms. FIGURE 3 shows the integration of the hierarchically structured FDIR modules (colored red) from system level FDIR to subsystem level FDIR (payload FD module, AOCS FD module, etc.) down to equipment level into the on-board software architecture. This architecture is based on a centralized on-board software data pool (OBSW DP). Also indicated is the event management module (Event Mgr.) that is triggered by system level FDIR.

3.4. Flight Control Procedures, Mission Time Line & On-Board Control Procedures

Traditional spacecraft operations are mainly done with Flight Control Procedures (FCPs) and the Mission Time Line (MTL). FCPs are executed step-by-step by a ground operator, which involves sending Telecommands (TC) to the spacecraft and checking Telemetry (TM) downlinked to ground. Missions with limited ground station coverage might also use the MTL, which is a sequence of time-tagged TC loaded from ground and executed by the on-board software when their time tag expires. As [16] and [17] outline, the MTL concept is limited as it does not allow for any logic while consisting in success-oriented commanding without immediate reaction to unexpected system behavior such as failed TC. Several articles [16-20], state that On-Board Control Procedures are not only appropriate for nominal operations handling; OBCPs are also suitable for more powerful on-board automation of spacecraft operations in order to allow the ground operators to prepare and uplink complex operations sequences. On-board control procedures are per the above established definition (see TAB 1) considered automated processes, not autonomous ones: OBCPs are activated via TC or triggered by listed events, no decision-making process on-board is involved.

On-board control procedures are software programs designed to be executed by an OBCP engine, which can easily be loaded, executed, and also replaced on-board the spacecraft. [12] These procedures are command scripts assembled from Packet Utilization Standard (PUS) service commands which are defined within [21]. As an example, PUS 8 implements functions e.g. for deployment of appendices. PUS 18 is reserved for on board functions that require more flexibility in either their execution flow or the function triggering parameter, i.e. OBCP loading from ground, execution control and parameter setting [13].
FIGURE 3 shows the FCP, MTL and OBCP operations concept. As can be seen, the OBCPs run in a dedicated, separate subsystem, i.e. in a safe "sandbox" environment.

Modern OBCPs are small, script-like programs written in a specific language called on-board control language (OCL), compiled to code and executed on-board in virtual machines. The language capabilities include features of high level languages like unsized arrays, loops, subroutines etc. Further details are found in [13], [12], [17], [18].

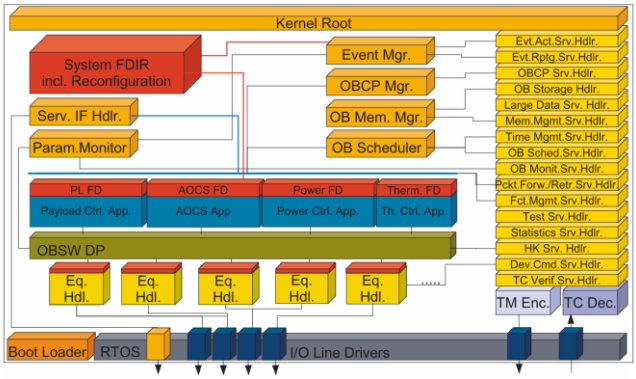

FIGURE 3. Failure detection, isolation and recovery organized in a hierarchical structure [13].

Several ESA missions apply the OBCP concept for nominal and FDIR operations, among others, there are to be mentioned the deep space missions Rosetta (launched 2004), Venus Express (launched 2005) [16] and the dual mission Herschel/Planck [17]. Mars Express (launched in 2003) was not foreseen to use many OBCPs until absolutely necessary from operational point of view due to system degradation in 2011 [20]. An improved implementation of OBCPs is reported in ESA's deep space missions Gaia (launch 2013) [13] and BepiColombo (launch 2015) [18].
Furthermore, the Earth observing missions CryoSat and GOCE apply a simplified concept and use on-board control procedures for implementation of Failure Detection, Isolation and Recovery algorithms (FDIR) [16].

Typical OBCP applications are scripts for equipment reconfiguration - e.g. switch over from nominal sensor to redundant [13]. In ESA's Rosetta mission, OBCPs are applied to payload mode changes, telemetry and telecommand (TT&C) system redundancy switching, power sharing in deep space, spacecraft warm-up after deep space hibernation mode, entry and exit from hibernation mode and support of FDIR functionalities such as establishment of safe mode in case of emergency [16], [19].

As depicted in FIGURE 4, the OBCPs run in a separated, safe environment. This architecture allows the creation and upload of new procedures in flight, while their update does not require modification, uplink and validation of the complete on-board software [17], [13]. Hence, OBCPs can be uploaded to spacecraft from ground any time and offer great flexibility, particularly when it comes to complex operational procedures that will be available only shortly before launch, e.g. separation sequence of BepiColombo [18].

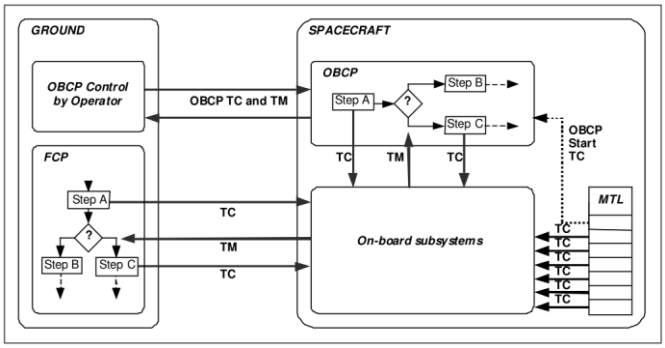

FIGURE 4. OBCPs vs. traditional spacecraft operations with FCPs and MTL [16].

Although the above description refers mainly to ESA spacecraft architectures, the spacecraft of the US National Aeronautics and Space Administration (NASA) use similar approaches: hardware redundancy, hierarchically distributed system and subsystem FDIR modules and a safe mode concept, refer for example to [22]. The New Horizons mission implemented an approach similar to ESA's OBCPs for spacecraft fault protection [11].

TAB 2. Fault detection and recovery on FDIR hierarchical levels. Isolation is often not performed [7], [13], [15].

Fault/Failure on	Impact	Fault Detection	System Recovery
Level 0	No impact on system performance	Local in unit internal unit checks, data transmission checks, consistency checks	Local in unit command retry, unit reinitialization, reboot
Level 1	Degraded performance of subsystem	in respective subsystem level the faulty unit belongs to limit checking of unit parameters, plausibility checks	By subsystem switch to redundant unit, command retry, reboot
Level 2	Performance loss of subsystem	Several alarms from unit level 0 consistency checks	Platform level switch to redundant side, command retry
Level 3		Faults on FDIR units	
Level 4	Performance loss of system, mission interruption	Several alarms from level 2 & 3, Hardware alarms e.g. Sun/Earth out of sensor view, thruster stuck open	Change operational mode to safe mode, wait for ground intervention to recover system and transfer system to nominal mode

4. GAP ANALYSIS

The conventional techniques currently used in space systems seem industrially well mastered and well characterized; many expected failures are anticipated and uncovered.
Hardware redundancy methods provide a high level of robustness and good performance, are easy to use and manage [8]. Main problems are increased mass and system complexity, hence, on low budget spacecraft there are other FDIR methods to be considered. An example provides [23] for NASA's WISE telescope mission.

Literature reports conventional FDIR methods suffering from significant shortcomings, like often missing isolation of faults and failures on-board [7], only partial observability of the actual system status and no on-board knowledge at all about the general operational capabilities of the system [9]. Selecting the thresholds for limit-checking techniques on unit/equipment level is a compromise between the detection size of deviations and the false alarm rate [24]. Often, the thresholds are chosen conservatively leading to a high mission outage times for many unnecessary safe mode events.

Mars Express, for example, lost half a year of operational time due to a SSMM failure that could repeatedly not be handled on-board, caused multiple safe mode transitions and operators needed more than six months to establish a suitable recovery action for the spacecraft [20]. The Earth observation mission CloudSat [25] suffered from severe degradation of batteries that conflicted with safe mode power requirements and caused extensive mission outage. Only the high effort of the ground operators to find a way around predefined, hardcoded routines in the satellites on-board software enabled the mission to be continued. A similar scenario was experienced by the Earth observation mission Cluster [26].
On the other hand, ESA's ADM-Aeolus mission for Earth observation claims having adopted from the beginning a high level of on-board autonomy: FDIR methods are implemented such, that nominal operations are resumed on-board after a single failure and thus, ground intervention is minimized. Operations are design to continue even in safe mode for five days without ground contact in failure case [27].

Further literature review revealed cases, where in extended mission phases operators were willing to implement on-board software features providing the spacecraft with a higher level of on-board autonomy. Possible explanations might be first, the acceptable risk for the spacecraft in such a late phase of the mission. Second, the increased knowledge about the spacecraft's operational behavior and system reactions might have led to higher confidence in foreseeing the system's behavior when implementing a higher degree of on-board autonomy. As an example, among others, the number of implemented OBCPs increased dramatically on ESA's Mars Express mission, when nominal operations could no longer be managed from ground [20].

The conclusion is drawn, that the extent of higher on-board autonomy with respect to system level FDIR implementation highly depends on mission needs. Determining factors are first, the kind of mission (Earth observation, communication, commercial or deep space mission) and second, the criticality of the considered mission phase. Implementations of intelligent FDIR methods on-board satellites will vary strongly with the mission scenarios and the expected benefits with respect to cost, performance and availability requirements. Particularly on highly complex interplanetary space missions in critical mission phases (orbit insertion maneuver, descent and landing, rendezvous and docking, atmospheric reentry etc.), advanced on-bard FDIR methods seem crucial in case of unexpected failure, because of long signal delay times that limit the ability of the ground crew, to respond to faults in a timely manner. This approach is enabled by today's rapid progress in computing power, one of the limiting factors in spacecraft design.

5. INNOVATIVE FDIR STRATEGIES

5.1. Analytical Redundancy

Analytical redundancy takes advantage of mathematical relations between measured and estimated variables resulting in nonzero discrepancies or so-called residuals and thus, faults or failures. Fault diagnosis is achieved by first, residual generation, then, residual evaluation and last, application of an appropriate decision logic. These set of methods is commonly called model-based, where models are understood as knowledge-based dynamic models (usually a set of differential equations in state-space form). Model-based fault diagnosis is considered a structured and mature field of research, many methods have been proposed and discussed in the control community. An overview can be obtained from standard text books [4, 28–30] and survey papers that focus on aerospace application [1, 2, 8].
Model-based FDI covers the areas of fault detection and fault isolation, i.e. from level 0 to 2, maybe up to level 3 of the hierarchical FDIR architecture of spacecraft (refer to TAB 2). This paper focuses on high level FDIR techniques, i.e. level 4. Thus, analytical redundancy is not investigated further.

5.2. Soft-Computing Methods

TAB 3 gives an overview of studies covering the last two decades with application of soft-computing or artificial intelligence methods in fault diagnosis of the (aero-)space domain. The studies are compared according to two criteria: type of study (level of detail of simulation, reported hardware tests) and reported application level (according to TAB 2). In addition, with respect to possible future spacecraft application, there should at least be considered the computational resources needed, suitability for real-time application and reliability (i.e. false alarm rate versus high detection rate). But since this information is hardly available, only the first two criteria apply within this paper.
Most of the cited studies are simulation based, some in great detail, others used simplified modeling of the system, environmental and boundary conditions. Outstanding studies are ESA's SMART-FDIR study [36] in 2003, that used the GOCE satellite simulation environment for validation purposes; NASA's Remote Agent Experiment (RAX) [34], that was tested in-flight as an on-board experiment in 1998 on the Deep Space 1 mission; and the work of [37], were a flight experiment of a Unmanned Aerial Vehicle (UAV) was conducted successfully.

TAB 3. Overview of studies investigating innovative fault diagnosis techniques, with references.

Analytical model based FDI	(Dynamic) Bayesian Networks	Neural Networks	Fuzzy Logic	Dempster-Shafer-Evidence Theory	Cognitive Automation
MEX thruster fault identification [24]	ESA's Advanced FDIR study [31]			Fault diagnosis [32]	Guidance of cooperative UAV [33]
Remote Agent Experiment (Livingstone) of Deep Space 1 [34]	Fault diagnosis in SSHM of small satellite [35]		SMART-FDIR [36]		Cognitive guidance of UAV [37]
	Fault diagnosis in power subsystem [38]		Fault diagnosis in spacecraft power subsystem [39]		Management of aircraft propulsion subsystem [40]
			Fault diagnosis in S/C reaction wheels [41]		
Fault diagnosis in GNC subsystem of reentry vehicle [42]	Mars rover system level FDIR [9]		Attitude and flight control for reentry vehicle X-38 [43, 44]		
			AOCS subsystem control [45]		
			AOCS subsystem (GEO station keeping, rendezvous & docking, instrument pointing) [46, 47]		
	Landing site selection [48]		Landing site selection [49]	Landing site selection [50]	

The majority of the above studies concentrate their fault detection efforts on component [32] or subsystem level (level 0 up to 3), e.g. [38, 39, 45], to decide which of the thrusters [24] or reaction wheels [41] is faulty. On subsystem level, studies focus on the power or attitude and orbit control subsystem (AOCS), which are indeed very essential for satellite operation and safety. But only few authors study beyond fault detection and isolation on subsystem level taking into account system recovery like ESA's SMART-FDIR study [36], or fault effect mitigation measures on system level (level 4), like the remote agent experiment [34]. The studies about guidance of a single or cooperative UAVs [33, 37], as well as the ARPHA study [9] of a rover power subsystem and respective influence on system level operations appear particularly suitable for the application on system level.

In summary, the cognitive automation approach following [51] seems to be promising for the development of future advanced FDIR methods. To the author's best knowledge, the cognitive automation approach is so far the only approach that is already tested successfully on actual flight hardware (except for the remote agent experiment). Further research revealed that cognitive methods are considered particularly suitable for intelligent behavior on system level [52]. They do support a centralized knowledge base to support situational awareness, what is stated a necessary improvement of spacecraft's system level FDIR [7, 9].

A future case study as described below will show, if cognitive automation is suitable for implementation in spacecraft system level FDIR.

6. RESEARCH CHALLENGES

Several issues have to be addressed, if the development of innovative methods for advanced system level FDIR methods is considered seriously.
Industry and operators need to build confidence into any newly introduced system level FDIR implementation. A suitable development roadmap leading to a flight test is mandatory to demonstrate a sufficient technology readiness level (TRL) of at least 7 [53]. One possibility to do so is outlined in the following.

6.1. Feasibility Study

Every development starts with what did the researchers of the studies cited above: a feasibility study based on use cases that require high complexity to prove the proposed concept, but need to be not too complex so the developer might still be able to understand what is going on during testing. Complexity of testing needs to be increased gradually, i.e. starting at a single subsystem model up to modeling and simulating all the satellite systems.

Successful simulation (TRL 3) shall be followed by building a prototype: the software needs to run on actual on-board computer hardware to prove that the implementation accounts for the usual constraints on-board spacecraft (i.e. limited computational load, memory, power and weight). Hardware-in-the-loop testing is the next step to introduce real faults and failures into the FDIR system and receive credible reactions, reaching TRL 4/5. The final step should be a flight test, as an experiment like RAX for a beginning, on a technology demonstrator satellite or on a commercial satellite in extended mission, that already

achieved all its mission objectives. Nanosatellites and cubesats do not seem to be an option, since their system complexity is usually low.

However, assumptions that are drawn from the above survey to be accounted for within the feasibility study are listed in the following. In non-critical mission phases, traditional methods are well-proven and are to be maintained. In critical mission phases, an interesting approach would be to assist the innovative FDIR methods of the system with traditional methods, integrated in the execution phase after decision-making process is completed. Of course, any new principle in system level FDIR complements, not substitutes, hardware redundancy and safe mode concepts.

6.2. Further Issues

The above mentioned implementation needs to prove its predictability, observability, performance and reliability.
A cost-benefit analysis needs to be conducted to provide evidence that the overall mission and operational cost and maintenance effort can be reduced when system safety and availability are increased at the same time.
A well-known research challenge with respect to artificial intelligence methods is the development of a suitable verification and validation process, an issue that needs to be addressed to go successfully through a qualification process for space application.
Finally, the effective integration of the development process of the innovative FDIR methods into the classical design phases of a satellite needs to be assessed.

7. CONCLUSION

Currently used FDIR techniques in the space domain appear very suitable in Earth observation applications, where human intervention in case of unexpected failure is possible several times a day. When it comes to commercial communication satellites, literature reports that a successful fail-safe approach has already been adopted.

In highly complex deep space missions, in case of unexpected failure in critical mission phases, improved FDIR methods meaning a higher level of on-board autonomy seem desirable. A literature review showed a high number of innovative fault diagnosis techniques that have yet been studied with application in the (aero-)space domain.

The cognitive automation approach seems very promising in this context for an application in the system level FDIR of spacecraft. Thus, a feasibility study will be conducted to provide some evidence if this assumption is valid.

8. ACKNOWLDEGMENTS

The authors would like to thank Prof. Axel Schulte from the Institute of System Dynamics and Flight Mechanics of the Bundeswehr University Munich for providing us with the Cognitive Systems Architecture (COSA) software package.

9. LITERATURE

[1] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter, “Model-based fault diagnosis for aerospace systems: a survey,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2012.

[2] D. Henry, S. Simani, and R. Patton, “Fault Detection and Diagnosis for Aeronautic and Aerospace Missions,” in *Fault tolerant flight control: A benchmark challenge*, C. Edwards, T. Lombaerts, and H. Smaili, Eds, Berlin: Springer Verlag, 2010, pp. 91–128.

[3] K. Patan, *Artificial neural networks for the modelling and fault diagnosis of technical processes*. Berlin: Springer, 2008.

[4] R. Patton, Ed, *Issues of fault diagnosis for dynamic systems*. London [u.a.]: Springer, 2000.

[5] W. Truszkowski, *Autonomous and autonomic systems: With applications to NASA intelligent spacecraft operations and exploration systems*. London: Springer, 2009.

[6] *Space engineering: Space segment operability*, ECSS-E-ST-70-11C, 2008.

[7] X. Olive, “FDI(R) for satellites: How to deal with high availability and robustness in the space domain?,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 99–107, 2012.

[8] A. Zolghadri, “Advanced model-based FDIR techniques for aerospace systems: Today challenges and opportunities,” *Progress in Aerospace Sciences,* http://www.sciencedirect.com/science/article/pii/S0376042112000292, 2012.

[9] D. Codetta-Raiteri, L. Portinale, S. Di Nolfo, and A. Guiotto, “ARPHA: a software prototype for fault detection, identification and recovery in autonomous spacecrafts,” *Acta Futura*, vol. 5, pp. 99–110, 2012.

[10] W. Fehse, *Automated rendezvous and docking of spacecraft*. Cambridge ;, New York: Cambridge University Press, 2003.

[11] R. C. Moore, “Autonomous safeing and fault protection for the New Horizons mission to Pluto: Bringing Space Closer to People, Selected Proceedings of the 57th IAF Congress, Valencia, Spain, 2-6 October, 2006,” *Acta Astronautica*, vol. 61, no. 1–6, pp. 398–405, http://www.sciencedirect.com/science/article/pii/S0094576507000604, 2007.

[12] *Space engineering: Spacecraft on-board control procedures*, ECSS-E-ST-70-01C, 2010.

[13] J. Eickhoff, *Onboard computers, onboard software and satellite operations: An introduction*. Berlin ;, New York: Springer, 2012.

[14] R. Gessner, B. Kosters, A. Hefler, R. Eilenberger, J Hartmann, and M Schmidt, “Hierarchical FDIR Concepts in S/C Systems,” in *Proceedings of the 8th International Conference on Space Operations (SpaceOps)*, Montreal, Canada, May. 2004.

[15] J. F. T. Bos, D. Zorita, A. Bacchetta, G. Chlewicki, D. Guichon, and I. Rasmussen, “ACMS FDIR System for the Herschel/Planck Satellites,” in *Proceedings of the 6th International ESA Conference on Guidance, Navigation and Control Systems*, 2006.

[16] C. Steiger, R. Furnell, and J. Morales, “OBSM Operations Automation through the use of On-

board Control Procedures," Montreal, Canada, 2004.

[17] M. Ferraguto, T. Wittrock, M. Barrenscheen, M. Paakko, and V. Sipinen, "The On-Board Control Procedures Subsystem for the Herschel and Planck Satellites," in *32nd Annual IEEE International Computer Software and Applications: COMPSAC '08*, 2008, pp. 1366–1371.

[18] A. Schwab, R. Eilenberger, and W. Zur Borg, "OBCPs - an integrated part of the BepiColombo Autonomy and Flexibility," in *The 12th International Conference on Space Operations: SpaceOps 2012*

[19] C. Steiger, R. Furnell, and J. Morales, "On-Board Control Procedures for ESA's Deep Space Missions Rosetta and Venus Express," in *Proceedings of DASIA 2005 - Data Systems in Aerospace*, Noordwijk: ESA Publ. Div, 2005.

[20] D. T. Lakey, M. Eiblmaier, M. Denis, B. de Teixeira Sousa, R. Porta, M. Shaw, and T. Francisco, "Multi-Mission End-to-End OBCP Configuration Control," in *The 12th International Conference on Space Operations: SpaceOps 2012*

[21] *Space engineering: Ground systems and operations — Telemetry and telecommand packet utilization*, ECSS- E- 70- 41A, 2003.

[22] P. S. Morgan, "Fault protection techniques in JPL Spacecraft," in *Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace (ISHEM)*, 2005.

[23] E. B. Rice and S. J. Lev-Tov, "Optimized Spacecraft Fault Protection for the WISE Mission," in *IEEE Aerospace Conference*, 2008, pp. 1–8.

[24] R. J. Patton, F. J. Uppal, S. Simani, and B. Polle, "Robust FDI applied to thruster faults of a satellite system," *Control Engineering Practice*, vol. 18, no. 9, pp. 1093–1109, http://www.sciencedirect.com/science/article/pii/S0967066109000707, 2010.

[25] M. Nayak, M. Witkowski, D. Vane, T. Livermore, M. Rokey, M. G. I. J. Barthuli, B. Pieper, A. Rodzinak, S. Silva, and P. Woznick, "CloudSat Anomly Recovery and Operational Lessons Learned," in *The 12th International Conference on Space Operations: SpaceOps 2012*

[26] I. Clerigo, S. Sangiorgo, and J. Volpp, "Avoiding Cluster Safe Modes," in *The 12th International Conference on Space Operations: SpaceOps 2012*

[27] K. Adamson, P. Bargellini, T. Nogueira, H. Nett, and C. Caspar, "ADM-AEOLUS: Autonomy, Automation, and Mission Planning Reuse," in *Space operations: Exploration, scientific utilization, and technology development*, C. A. Cruzen, J. M. Gunn, and P. J. Amadieu, Eds, Reston: Published by the American Institute of Aeronautics and Astronautics, 2011, pp. 341–358.

[28] R. Isermann, *Fault-diagnosis applications: Model-based condition monitoring -- actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Heidelberg ;, New York: Springer Verlag, 2011.

[29] M. Witczak, *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches,* 1st ed. Berlin: Springer Berlin, 2007.

[30] S. X. Ding, *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools*. Berlin: Springer, 2008.

[31] N. Holsti and M. Paakko, "Towards advanced FDIR components," in *DASIA: Data Systems in Aerospace*, Noordwijk: ESA Publ. Division, 2001.

[32] Y. Wu, Z. Ren, and Z. Zeng, "Fault diagnosis method based on D-S evidence theory," in *Prognostics and Health Management Conference: PHM '10*, 2010, pp. 1–4.

[33] C. Meitinger, *Kognitive Automation zur kooperativen UAV-Flugführung*. Dissertation am Institut für Systemdynamik und Flugmechanik der Universität der Bundeswehr München, Neubiberg, 2008.

[34] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams, "Remote Agent: to boldly go where no AI system has gone before," Artificial Intelligence, vol. 103, no. 1–2, pp. 5–47, 1998.

[35] J. Schumann, O. Mengshoel, and T. Mbaya, "Integrated Software and Sensor Health Management for Small Spacecraft," in *Space Mission Challenges for Information Technology (SMC-IT), 2011 IEEE Fourth International Conference on*, 2011, pp. 77–84.

[36] A. Guiotto, A. Martelli, and C. Paccagnini, "SMART-FDIR: Use of Artificial Intelligence in the Implementation of a Satellite FDIR," in *DASIA: Data Systems in Aerospace*, 2003.

[37] M. Kriegel, *Wissensbasierte Konfiguration eines unbemannten Fluggeräts als Architekturansatz zur kognitiven Flugführung*. Dissertation am Institut für Systemdynamik und Flugmechanik der Universität der Bundeswehr München, Neubiberg, 2012.

[38] B. Ricks and O. J. Mengshoel, "Methods for Probabilistic Fault Diagnosis: Electrical Power System Case Study," Silicon Valley Campus, Paper 60, 2009.

[39] D. Cayrac, D. Dubois, and H. Prade, "Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application: Fuzzy Systems, IEEE Transactions on," *Fuzzy Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 251–269, 1996.

[40] W. Pecher, S. Brüggenwirth, and A. Schulte, "Using cognitive automation for aircraft general systems management," in *SoSE: 5th International Conference on System of Systems Engineering*, 2010, pp. 1–8.

[41] N. Meskin and K. Khorasani, *Fault detection and isolation: Multi-vehicle unmanned systems*. New York: Springer, 2011.

[42] A. Falcoz, D. Henry, and A. Zolghadri, "Robust Fault Diagnosis for Atmospheric Reentry Vehicles: A Case Study," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 5, pp. 886–899, 2010.

[43] S. Wu, "Fuzzy logic based attitude control of the spacecraft X-38 along a nominal re-entry trajectory," *Control Engineering Practice*, vol. 9, no. 7, pp. 699–707, 2001.

[44] "Fuzzy logic based full-envelope autonomous flight control for an atmospheric re-entry spacecraft," *Control Engineering Practice*, vol. 11, no. 1, pp. 11–25, 2003.

[45] F. Zotes and M. Peñas, "Intelligent satellites control based on fuzzy logic in the Earth-Moon Libration points," in *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, 2010, pp. 605–610.

[46] G. Ortega and J. Giron-Sierra, "Fuzzy logic techniques for intelligent spacecraft control systems,"

in *IEEE International Conference on Systems, Man and Cybernetics: Intelligent Systems for the 21st Century*, 1995, pp. 2460 -2465 vol.3.

[47] G. Ortega, A. J. Mulder, and H. Verbruggen, “Fuzzy Logic for Spacecraft Control: A European Approach,” in *Artificial Intelligence, Robotics and Automation in Space*, 1999, pp. 471–476.

[48] N. Serrano, “A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent,” in *IEEE/RSJ: International Conference on Intelligent Robots and Systems*, 2006, pp. 5112–5117.

[49] S. Ploen, H. Seraji, and C. Kinney, “Determination of Spacecraft Landing Footprint for Safe Planetary Landing,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 1, pp. 3–16, 2009.

[50] H. Seraji and N. Serrano, “A Multisensor Decision Fusion System for Terrain Safety Assessment,” *Robotics, IEEE Transactions on*, vol. 25, no. 1, pp. 99–108, 2009.

[51] R. Onken and A. Schulte, *System-ergonomic design of cognitive automation: Dual-mode cognitive design of vehicle guidance and control work systems*. Berlin, Heidelberg: Springer-Verlag, 2010.

[52] P. Langley, J. E. Laird, and S. Rogers, “Cognitive architectures: Research issues and challenges,” *Cognitive Systems Research*, vol. 10, no. 2, pp. 141–160, http://www.sciencedirect.com/science/article/pii/S1389041708000557, 2009.

[53] J. C. Mankins, *Technology Readiness Levels: A White Paper.* Available: http://www.dsto.defence.gov.au/attachments/TRLs%20A%20White%20Paper.pdf (2012, Aug. 08).