FDIR Handling in Eu:CROPIS

Olaf Maibaum (1), Ansgar Heidecker (2), Fabian Greif (2),
Markus Schlotterer (2), Andreas Gerndt (1)

(1) German Aerospace Center (DLR), Simulation and Software Technology,
Lilienthalplatz 7, 38108 Braunschweig, Germany

(2) German Aerospace Center (DLR), Institute of Space Systems,
Robert Hooke Str. 7, 28359 Bremen, Germany

ABSTRACT: Fault detection, isolation, and recovery (FDIR) mechanisms in on-board software are essential to guarantee the survival of the satellite in case of a hardware malfunction. E.g., outage of essential attitude control system (ACS) actuators or sensors can lead to mission loss. The on-board software has to handle such situation autonomously by switching to cold redundant devices or by isolation of information from hot redundant devices. The FDIR implementation for the ACS of the spin stabilized small satellite Eu:CROPIS (Euglena Combined Regenerative Organic food Production In Space) is shown in this paper.

1. INTRODUCTION

This paper shows the implemented software architecture and methods for FDIR handling in the Eu:CROPIS ACS. It starts with a short introduction of the Eu:CROPIS mission and the implemented ACS. Sections 3 and 4 show the underlying execution platform and the used services from the Package Utilization Service (PUS) standard. The software architecture and FDIR handling is presented in section 5. The paper closes with the conclusion and outlook on future work.

2. EU:CROPIS

The mission Eu:CROPIS is the demonstration of the feasibility of restartable and sustainable life support systems on a pure biological basis at target gravities 0.16 g (Moon) and 0.38 g (Mars). Such systems enable the production of food and atmosphere from waste like urine and phosphate [1]. The launch of the mission was at 03.12.2018 as part of Spacefligth's SSO-A rideshare mission launched from Vandenberg Air Force Base with a Falcon 9 launch vehicle.

The main requirement to be fulfilled by the ACS is to generate gravity in the biological compartments and to orient the z-axis into sun direction. The satellite bus is spinning around the z-axis between 5 to 31 rpm. To keep sun orientation of solar panels, the spin axis has to be reoriented by ~1 deg/day.

As actuators, the ACS uses three magnetic torquers to control the rotation and spin axis. It uses as sensors two magnetometers, ten sun sensors and four angular rate gyroscopes. Sensor data is filtered by an Unscented Kalman Filter (UKF) which is designed for spin stabilized satellite. For a detailed description of the used UKF and the ACS, see [2].

3. TASKING FRAMEWORK

The communication and processing inside the Eu:CROPIS ACS is realized by the Tasking Framework. It is a reactive asynchronous execution platform with support for multi-core processors and extensions for distributed on-board systems. For the Eu:CROPIS ACS, a non-distributed single core configuration is used. The mean for communication in the Tasking Framework are channels, specialized to different types of data buffers by overwriting the basic channel class. A task with its computations is started by the Tasking Framework when on all incoming channels of the task the expected amount of data is published. More details of the Tasking Framework can be found in [3].

4. PUS SERVICES

The implemented services in the Eu:CROPIS command and data handling system (CDH) follow the PUS standard. [4] Each application of a subsystem has to register service handler at the CDH system to provide the service implementation of the subsystem. By the application and service identification of a PUS data package, the CDH system can call the subsystem specific implementation of a service request.

- *Device Command Distribution Service*: Accesses parameters of the ACS. When implementation was started this service was used in lack of the *Parameter Management Service* which is now available in the updated PUS standard. [4]
- *Diagnostic Reporting Service*: Provides regular housekeeping parameter reports and diagnostic parameter reports. A diagnostic parameter report can address data from all internal ACS channels with up to 10 Hz or immediately if the filter matches, e.g. on equality to an error identification.
- *Event Reporting Service*: Announces events in the system. The service is implemented by the CDH subsystem with a topic interface. A topic is a synchronous communication interface, provided by the used outpost-core library. [6]
- *Function Management Service*: Provide means to initiate special functionalities in the software

5. FAILURE HANDLING

5.1 Software Architecture

The whole software architecture of the Eu:CROPIS ACS is based on the Tasking Framework [3]. The software architecture concepts for the implementation of hardware interfaces and the UKF are described in [7].

The FDIR handling in the Eu:CROPIS ACS has several logical layers. In the first logical layer, software interfaces to actuator and sensor hardware and the UKF detect and isolate failures. All failures are reported by their error identification to the error channel.

The error channel triggers two tasks. The error filter task sends diagnostic data if the error identification matches to one of the predefined FDIR diagnostic data reports, which contain all internal data channels with a relation to the failure. An upper limit of sent reports can be specified by a parameter to prevent flooding the telemetry memory by a permanent failure. The error handle task increases the error counter which corresponds to the error identification. If an error counter is above a limit, the error handle task looks into the mapping of critical event reports and if one is defined for the error counter, it is send by the event reporting service to the system, which initiates a recovery action at system level. The limit can be configured by a parameter.

Figure 1 shows all related software components, tasks, and channels responsible for FDIR handling in the Eu:CROPIS ACS.

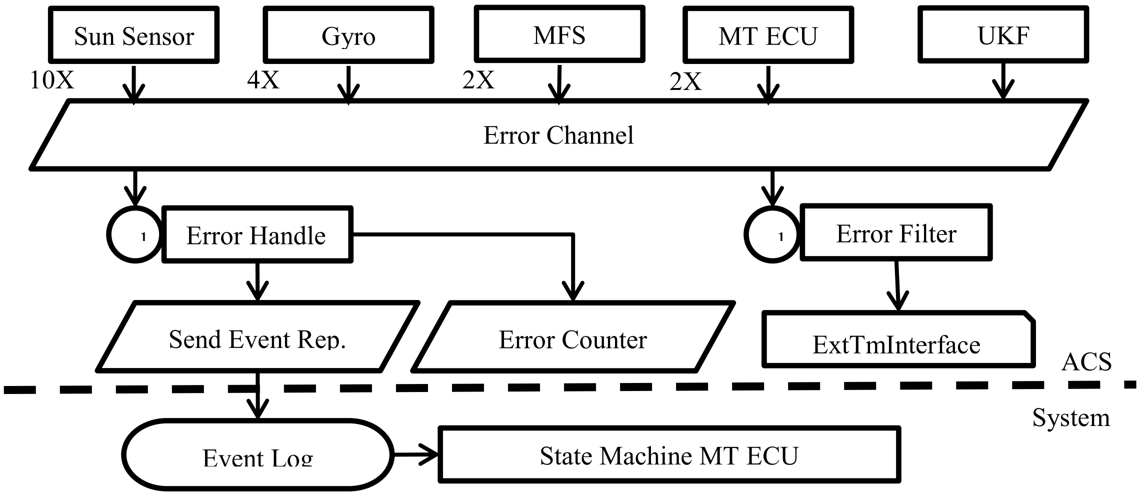

Figure 1 FDIR Software Architecture in the Eu:CROPIS ACS

5.2 Detection and Isolation

Failure detection is executed by the software interfaces to the sensor and actuator hardware. At first, the communication with the sensor and actuator hardware is checked. These checks are: check on successful sending of messages to the hardware; successful receiving of messages from the hardware; and format and data of received messages is valid. At second, the status information in the message data is interpreted. On basis of the detected state, no data will sent or the data is sent with an attached quality metric.

The second step in failure detection is performed by the UKF. First, it checks sensor data on validity, e.g. measured sun angle data during eclipse is invalid. For all valid sensor data, the Mahalanobis distance is computed to detect inconsistent state information from a sensor device. Implausible data will be isolated from the computation of the high accurate attitude state information which is sent to the controller.

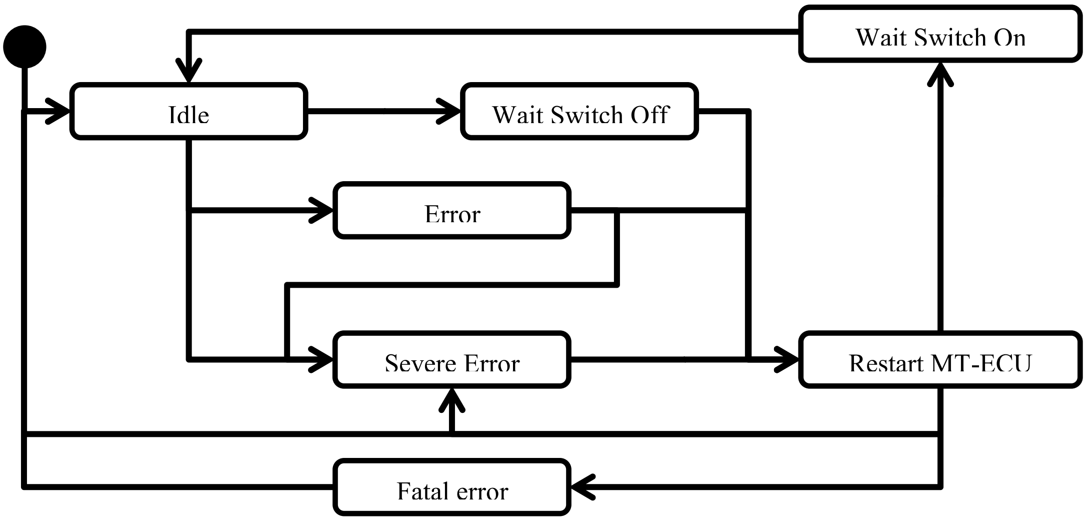

Figure 2 State Machine for Recovery Actions of Magnetic Torquer

5.3 Recovery

Reported critical failure events are managed by the system state machine and can initiate a power reset or a switch to cold redundant hardware devices. Figure 2 shows the state machine for recovery actions of the MT-ECU (Magnetic-Torquer Electronic-Control-Unit). The state machine guarantees that two MT-ECUs are not running at the same time by the three states “Wait Switch Off”, “Restart MT-ECU”, and “Wait Switch On”. The state machine has three error states. In state “Error”, a power cycle on the MT-ECU is initiated. In state “Severe Error”, a switch to the cold redundant MT-ECU is initiated. In state “Fatal Error”, no MT-ECU is left for operation and the state machine waits on further action from ground operation.

6. CONCLUSIONS AND OUTLOOK

The paper shows the architecture and means for the FDIR handling of the Eu:CROPIS ACS. First checks for failure detection on communication status and data from the actuators are applied at interface level. At the UKF, the Mahalanobi distance is used to isolate implausible data. Each failure detection is reported to two tasks in the ACS, one to send diagnostic data, and the other to count errors and to initiate the recovery actions. On system level, a state machine is used to manage the power state of devices. First experiences in the LEOP and commissioning of the Eu:CROPIS mission show that the implemented FDIR mechanism works as expected. The reasoning for one increasing FDIR counter on the flight model in space can be analyzed by the diagnostic reporting service itself and the trigger by FDIR events. Additional, we get a full insight of all internal ACS data with a 10 Hz resolution by the diagnostic reporting service. All applied services for FDIR handling can be reused as configured in future mission.

7. REFERENCES

[1] S. Kottmeyer, C.F. Hobbie, F. Orlowski-Feldhusen, F. Nohka, T. Delovski, G. Morfill. The Eu:CROPIS Assembly, Integration and Verification Campaigns: Building the first DLR Compact Satellite. IAC 2018, Bremen (2018).

[2] A. Heidecker, T. Kato, O. Maibaum, M. Hölzel, Attitude Control System of the Eu:CROPIS Mission. IAC 2014, Toronto, Canada (2014).

[3] Software Evolution from TET-1 to Eu:CROPIS. 10th IAA Symposium on Small Satellite for Earth Observation. pp. 195-198. Berlin (2015)

[4] ECSS-E-70-41A: Ground Systems and Operations – Telemetry and Telecommand Packet Utilization, 30 January 2003.

[5] ECSS-E-ST-70-41C: Telemetry and Telecommand Packet Utilization, 15. April 2016.

[6] https://github.com/DLR-RY/outpost-core: Outpost Core. Link checked at 13. December 2018.

[7] O. Maibaum, A. Heidecker. Software Evolution from TET-1 to Eu:CROPIS. In: Digest of the 10th International Symposium of the International Academy of Astronautics. pp. 195-198. Wissenschaft und Technik Verlag, Berlin (2015)