

SpaceOps-2023, ID # 481

### New Safe Mode for INTEGRAL Mission

**G. De Marco<sup>a\*</sup>, J. Martin<sup>b</sup>, L. Toma<sup>c</sup>, T. Godard<sup>d</sup>, R. Southworth<sup>e</sup>, S. De Padova<sup>f</sup>, M. Baab<sup>g</sup>, A. Magunia<sup>h</sup>,  
D. Salt<sup>i</sup>, P. Chapman<sup>j</sup>, T. Strandberg<sup>k</sup>, M. A. Hann<sup>l</sup>, O.J. Page<sup>m</sup>, A. J. Kirkham<sup>n</sup>, S. Carnochan<sup>o</sup>.**

<sup>a</sup> RHEA for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße Darmstadt, Germany, [greta.de.marco@ext.esa.int](mailto:greta.de.marco@ext.esa.int)

<sup>b</sup> ESA/ESOC, OPS-OAI, Flight Control team, Robert-Bosch Straße 5, Darmstadt, Germany, [jim.martin@esa.int](mailto:jim.martin@esa.int)

<sup>c</sup> TelespazioVega for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße, Darmstadt, Germany, [liviu.toma@ext.esa.int](mailto:liviu.toma@ext.esa.int)

<sup>d</sup> RHEA for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße, Darmstadt, Germany, [thomas.godard@ext.esa.int](mailto:thomas.godard@ext.esa.int)

<sup>e</sup> ESA/ESOC, OPS-OAI, Flight Control team, Robert-Bosch Straße 5, Darmstadt, Germany, [richard.southworth@esa.int](mailto:richard.southworth@esa.int)

<sup>f</sup> TelespazioVega for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße, Darmstadt, Germany, [stefano.de.padova@ext.esa.int](mailto:stefano.de.padova@ext.esa.int)

<sup>g</sup> TelespazioVega for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße, Darmstadt, Germany, [marius.baab@ext.esa.int](mailto:marius.baab@ext.esa.int)

<sup>h</sup> TERMA GmbH for European Space Operations Centre (ESA/ESOC, OPS-OAI), Flight Control team, Robert-Bosch Straße, Darmstadt, Germany, [arnfried.magunia@ext.esa.int](mailto:arnfried.magunia@ext.esa.int)

<sup>i</sup> [djsalt55@gmail.com](mailto:djsalt55@gmail.com)

<sup>j</sup> AIRBUS DEFENSE AND SPACE Ltd, Gunnels Wood Road, SG1 2AS, Stevenage, UK, [patrick.chapman@airbus.com](mailto:patrick.chapman@airbus.com)

<sup>k</sup> Space Tech GmbH Seelbachstr. 13, 88090 Immenstaad, Germany contracting to AIRBUS DEFENSE AND SPACE GmbH, [tommy\\_strandberg@web.de](mailto:tommy_strandberg@web.de)

<sup>l</sup> CGI IT UK Ltd, Clothier Road, Bristol, BS4 5SS, UK, [mark.hann@cgi.com](mailto:mark.hann@cgi.com)

<sup>m</sup> CGI IT UK Ltd, Clothier Road, Bristol, BS4 5SS, UK, [olly.page@cgi.com](mailto:olly.page@cgi.com)

<sup>n</sup> CGI IT UK Ltd, Clothier Road, Bristol, BS4 5SS, UK, [andrew.kirkham@cgi.com](mailto:andrew.kirkham@cgi.com)

<sup>o</sup> CGI IT UK Ltd, Clothier Road, Bristol, BS4 5SS, UK, [stuart.carnochan@cgi.com](mailto:stuart.carnochan@cgi.com)

\* Corresponding Author

### Abstract

The International Gamma-Ray Astrophysics Laboratory (INTEGRAL) is a European space observatory mission that can observe objects in gamma/X-ray and visible wavelengths. The spacecraft was launched in 2002 and it is currently in its extended mission phase.

In the past two years, INTEGRAL underwent two major anomalies, which have dramatically changed the way the spacecraft is operated. After the first one, the thrusters were considered unreliable to perform reaction wheels (RW) angular momentum dumping. As alternative solution, the external perturbations due to the solar radiation pressure and the Earth gravity gradient were exploited to keep RW angular momentum under control. However, it was believed that the satellite would be capable of supporting one or two thruster-based safe mode (called ESAM, Emergency Safe Attitude Mode).

The second anomaly, so-called ESAM#9 (the 9<sup>th</sup> ESAM since launch), resulted in high spacecraft rates and intermittent loss of Sun illumination on the body-fixed solar arrays. ESAM#9 highlighted the need of a thrusterless reaction, relying only RWs, to recover any future loss of attitude control.

For this purpose, the INTEGRAL Flight Control Team (FCT), together with industry, is currently designing an on-board software patch which will bring an entirely new safe mode (NSM) to life. At first the NSM has been prototyped and deployed with ground-based automation: the strategy consists of first bringing the satellite rates down to zero before re-orienting the spacecraft into a sun-facing attitude assuring that enough power is generated on board from the solar panels and at the same time respecting instruments heating constraints. This is done commanding directly the Reaction Wheels angular momentum. Operationally INTEGRAL uses three wheels drive control law for attitude management, but the baseline for the NSM is to use four wheels.

Unfortunately, ground implementation implies continuous and stable coverage. Therefore the aim is to deploy the NSM as on-board software patch. Originally 1.7Kwords of available space in the on-board computer was estimated,

however an analysis ran by industries proved that there were 24K words of unmapped and empty memory which would be the target for the on-board NSM. Access and usage to this additional memory area has been proved with the RF-Suitcase.

The NSM control law has been prototyped and tested using JavaScript together with the operational simulator. It is comprehensive of three different control law types: the first consists in different fix-rate/one axis rotations drift proposals based on Sun position, the second implements a two-axis Sun position based proportional control law, and the third one takes care of scaling down the requested RW angular momentum in case any of the wheels is saturated. The development will follow a phased approach in which the control laws will be implemented sequentially.

The JavaScript prototype has been tested with different failure scenarios and it successfully passed the Preliminary Design Review. It was translated by industry in ADA code and uploaded into the on-board computer on the simulator. A complete validation plan has been executed before deployment and commissioning on the spacecraft.

This paper will detail all above mentioned design and implementation and will present status and future development.

**Keywords:** INTEGRAL, Safe Mode, Control Algorithm, Gamma-rays, patch.

### Nomenclature

A	Reaction Wheels orientation matrix in spacecraft reference frame
h	Angular momentum, [Nms]
w	Angular rate, [as/s]

### Acronyms/Abbreviations

ACC	Attitude Control Computer
ADA	Programming language named after Ada Lovelace, mathematician and the first computer programmer
AOCS	Attitude and Orbit Control System
ARO	Automatic Reconfiguration Order
CDMU	Command and Data Management Unit
ESAM	Emergency Safe Attitude Mode
FCE	Failure Control Electronics
FCT	Flight Control Team
FDE	Failure Detection Electronics
FDCE	Failure Detection and Control Electronics
FDIR	Failure Detection Isolation Recovery
FoV	Field of View
FSS	Fine Sun Sensors
IMU	Inertial Measurement Unit
IPS	Inertial Pointing and Slew
MOC	Mission Operation Control
NSM	New Safe Mode
RACP	Re-Action Control Procedures
RMU	Rate Measurement Unit
RW	Reaction Wheels
RWB	Reaction Wheel Bias
SAS	Solar Acquisition Sensor
SC	Spacecraft
SEU	Single Event Upset
SRP	Solar Radiation Pressure

## 1. Introduction

The International Gamma-Ray Astrophysics Laboratory (INTEGRAL) is operated by ESA as part of its Horizon 2000 program, [1]. INTEGRAL is dedicated to the fine spectroscopy and imaging in the Gamma- and X-rays high energy regime. The goal of this mission is to study violent events in the universe like black holes, neutron stars, active

galactic nuclei and supernovae. In recent years, INTEGRAL's data also contributed to the confirmation and study of gravitational waves. Its payload is composed by SPI [2], the Gamma-rays spectrometer, IBIS [3], the Gamma-rays imager, two JEM-X [4], the X-rays spectrometers and one optical monitor, OMC [5]. The observing direction is given by the spacecraft +X body axes, while solar arrays are perpendicular to the +Z body axes and are body fixed (Fig. 1).

Currently INTEGRAL highly elliptical orbit has a period of almost 3 days, and, due to its shape, the spacecraft is visible almost constantly from the network of ground stations, apart from a few hours around perigee. Therefore, the platform is equipped with only limited onboard autonomy and operations are almost completely conducted real-time from the European Space Operation Center (ESOC, in Darmstadt, Germany). The science center is located near Madrid, Spain at the European Space Astronomy Center (ESAC).

INTEGRAL was launched in October 2002 on a Proton rocket and was originally planned for a mission lifetime of less than a decade but operates without major degradation and is facing a very high scientific demand. As a matter of fact, until 2020, INTEGRAL had a fully functioning thruster-based Emergency Attitude Safe Mode, ESAM. There have been 9 ESAM since launch. In 2020 and in 2021, the spacecraft experienced to two major anomalies which lead to the disabling of ESAM and therefore the need to design a New Safe Mode, NSM. Outcome of the first anomaly (section 2), was that the propulsion subsystem was considered unreliable and therefore it could not be used anymore to de-saturate Reaction Wheels (RWs). However, it was thought that the thrusters could still support 1-2 safe modes. The second anomaly (section 3.2), ESAM number 9, made clear instead that the old ESAM cannot be trusted anymore and from here the need to design a New Safe Mode (section 4). The NSM is based on the recovery manually performed to save INTEGRAL mission during ESAM9. The NSM software patch has been first prototyped in JavaScript and tested with the operational simulator, and then translated in ADA (language used by the on-board software). All tests performed has been successful (section 4.3) and the patch is currently ready for deployment.

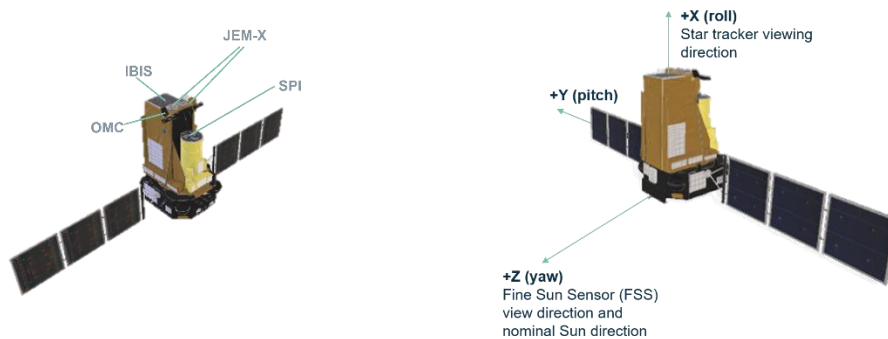


Fig. 1. INTEGRAL schematic representation.

## 2. Thruster-less operations for INTEGRAL Mission

### 2.1 Emergency Attitude Safe Mode Number 8: the first propulsion subsystem anomaly

On 16 May 2020, during RW desaturation activities, a severe under-performance of one thruster induced an ESAM (the 8th since launch), [6]. Whilst still in ESAM, the satellite unexpectedly de-pointed about 75° from the safe attitude, simulations showed that the ESAM controller could cause such a de-pointing in case of significant thruster under-performance. During this phase, a 5% drop of the propellant pressure was observed. The anomalous behavior suggested that nitrogen (i.e., pressurant) permeation through the tank membrane had also caused bubbles of gaseous nitrogen to form on the propellant side [7], as shown in Fig. 2.

Indeed, RW desaturation performed after ESAM8 led to a gradual decrease of the pressure inside the propulsion system. Two months after ESAM8, on July 17<sup>th</sup> 2020, a loss of 3 bars was observed. By then a new way to manage the accumulated RW angular momentum had been identified and was ready for in-flight validation, the so-called Z-Flip, subject of section 2.2.

### 2.2 Z-flip Operations

Following the major propulsion subsystem described introduced above, the management of the external torques (induced mainly by the solar radiation pressure) that had been routinely performed until that moment as per mission design with reaction wheel biases, was not considered safe and reliable anymore due to the irregular and unpredictable thruster firing and residual propellant uncertainty. The new strategy does not rely on thrusters anymore and takes advantage of the solar radiation pressure to keep the spacecraft total angular momentum under control.

The basic concept on which Z-flip is based is the following. Observing one side of the galactic plane, the spacecraft accumulates angular momentum in one direction. Midway through the revolution, a 180° slew around yaw/sunline is commanded. Now observing the opposite side of the galactic plane, the solar radiation pressure from this time onward builds up and induces an angular momentum opposite to the one accumulated during the first part of the revolution. The total angular momentum is conserved but it gets transferred between wheels. Therefore, it is possible to control the reaction wheels angular momentum through selection of targets in such a way to have a suitable yaw slew pattern (Z-flip manoeuvres).

INTEGRAL Science Operation Centre selects targets sequence where the angular separation is such that the XY-angular momentum accumulated by the RW increases and decreases in equal measure over the mission planning period under exam (1-2 revolutions). Z-flip is not actually performed at a specific time in the revolution, but the target selection sequence needs to consider the angular momentum accumulated by the wheels in such a way to select target from region of the sky to compensate this effect. INTEGRAL rotates about the Sun line over a planning period, stopping at intermediate targets along the way. The full range of observing attitudes is still available, but the order in which the observations are executed has become more crucial.

The Z-angular momentum is only mostly influenced by the gravitational perturbations at perigee.

A simplified representation of the Z-flip is shown in Fig. 3. In the first part of the revolution the angular momentum around X and Y is built up (graph on the bottom, XY-angular momentum is given by the normalised X- and Y-angular momentums). Then, the Z-flip manoeuvre is performed which brings the XY-angular momentum down again (in blue in Fig. 3). The image shows the Z-angular momentum which has a constant trend: only close to perigee the gravity gradient has an influence on the Z-angular momentum. To control the Z-angular momentum, specific attitudes are performed for perigee passage.

Z-Flip is now considered as baseline in mission planning activities, and it was demonstrated that this strategy does not impact the usable science time.

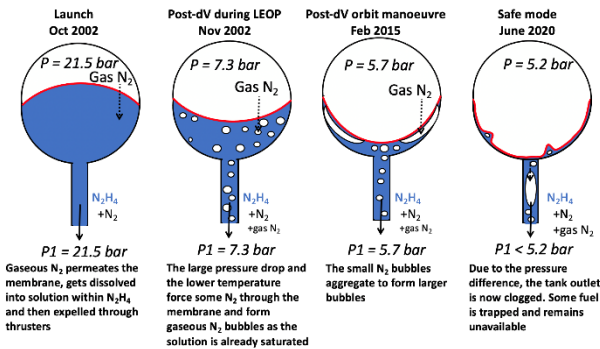


Fig. 2. Permeation of nitrogen through the tank membrane over the mission lifetime, [7].

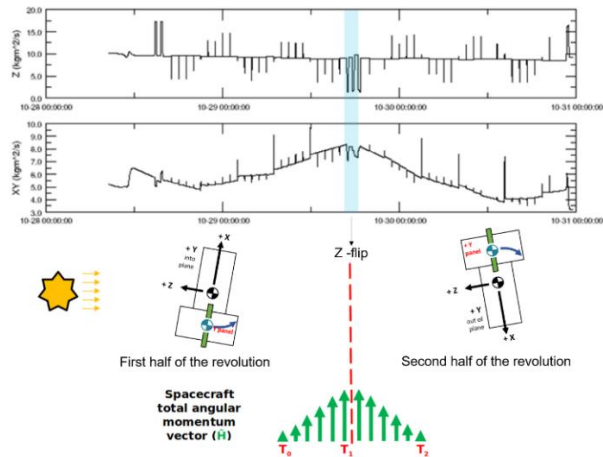


Fig. 3. Z-Flip simplified example.

### 3. Integral Emergency Attitude Safe Mode Number 9

In 2020, as outcome of the thruster anomaly investigation, it was determined that the fuel remaining in the pipes could have been enough to support at least another safe mode (i.e., ESAM) due to some propellant left in the thruster pipes [7]. On 22nd September, 2021, this assumption turned out to be wrong when the switch off of the active RWs, caused by a Single Event Upset (SEU), triggering the Emergency Attitude Safe Mode number 9 (ESAM9, section 3.2), during which the attitude could not be kept under control due to thruster underperformances.

#### 3.1 The Emergency Attitude Safe Mode (ESAM) concept

In nominal cases the INTEGRAL attitude is managed by the Attitude Control Computer, ACC, part of the Attitude and Orbit Control Subsystem (AOCS). The spacecraft is also equipped with the FDCE unit, Failure Detection and Control Electronics, which function is similar to what in more modern spacecraft is identified as FDIR. The FDCE unit is composed of two functional elements: the detection (FDE) and the control (FCE) parts. The FDE monitors different attitude sensors readings: if one of the constraints defined is exceeded then the Automatic Reconfiguration

Order, ARO, signal is generated. This instructs the spacecraft to take the ACC out of the control loop and the attitude control handling is handed over to the FCE. This automatically commands the reconfiguration of the spacecraft and start the search of a Sun pointing attitude to ensure a solar panel exposition towards the Sun: this is achieved using thrusters. As already introduced, this mode is the so-called ESAM and uses the Solar Acquisition Sensors (SASs) for attitude determination, Rate Measurement Units (RMUs) for rate information and thrusters for actuation. Designed with a cubic shape, SAS1, SAS2 and SAS3 mounted on three different panels of the service model, respectively pointing toward to +X, +Y and -Y directions are enough to determine the spacecraft attitude in case of emergency. Having a field of view of +/-97deg, an orientation of the SAS ensures a certain level of overlap and minimise the blind spot of the back side of the service module.

### 3.2 ESAM Number 9

On 22 September 2021, one of the active RWs (nominally INTEGRAL uses only 3 out of the 4 RWs for attitude control, i.e., RW2, RW3 and RW4) was switched off due to a Single Event Upset (SEU), [8]. The angular momentum stored in this wheel (RW3) was transferred into spacecraft rotation. This can be observed in the leftmost part of Fig. 4: a little spike is visible in the spacecraft rates while the spacecraft body was absorbing RW3 angular momentum.

The loss of attitude was detected by the FDE (Failure Detection Electronics), ARO signal was triggered, and the thrusters fired in the attempt to keep the solar panel Sun-faced (ESAM9 entry). ESAM turns off the IMUs therefore rates readings go to 0 as/s in Fig. 4 few minutes after the SEU.

Unfortunately, the thrust generated by the propulsion system was not sufficient to maintain a stable attitude and the spacecraft kept spinning at a rate of about 16°/minute (1000as/sec) on at least two axes (Fig. 4, i.e., roll and pitch), and the wheels saturated (+/- 4000 rpm), [7]. By the time the team assessed the situation and started the recovery and powered on the IMUs, the spacecraft had already lost the Sun illumination while continuing to tumble. The telemetry link was intermittent, coming in for short periods of time. Batteries were discharging, with some charging when the panels briefly faced the Sun.

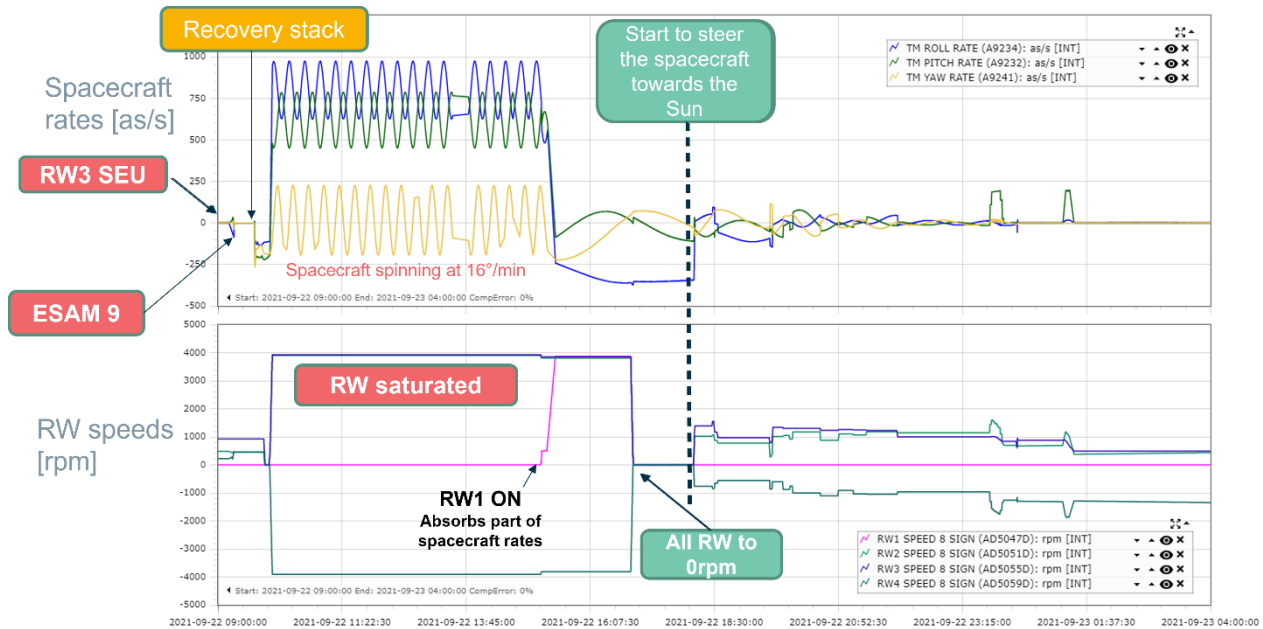


Fig. 4. INTEGRAL rates and Reaction Wheels speeds during Emergency Attitude Safe Mode number 9.

The first estimate of the batteries power before the potential loss of the satellite suggested a remaining lifetime 3 hours. Therefore, the entire payload was switched off, extending the recovery window by another 2 to 3 hours. To stop the spacecraft from spinning without the help of the thrusters, the spare RW (RW1) was activated (Fig. 4). RW1 absorbed part of the spacecraft body rates as shown in the leftmost side of Fig. 4. Even if the spacecraft body rates were significantly decreased, they were not yet down to 0 as/s and the spacecraft was not constantly facing the Sun, i.e., batteries were still discharging with brief recharging intervals.

The ESAM9 recovery actions, aimed to reach a Sun pointing attitude with zero residual body rates. First all RW speeds were commanded to 0rpm (Fig. 4), then, knowing the spacecraft and RWs inertias and rates, the team estimated, in real-time, the amount of angular momentum to be commanded to each RW in order to rotate the satellite to bring the sunlight back on the solar panels allowing batteries to recharge. After several manoeuvres, the attitude control was eventually regained: the Sun appeared on SAS1 (perpendicular to +Z axis which is the Sun pointing axis, Fig. 1) and further slews were performed to reach optimal Sun-angle (i.e., roll within +/-5° and pitch within +/-40°), as dictated by instruments thermal constraints and illumination of the solar panels. Recovery had been performed using three of the four RWs. Immediately after the anomaly, ESAM was disabled leaving the satellite without any Emergency Safe Attitude Mode, at least until December 2022, when a preliminary version of the new safe mode, NSM, was implemented using autonomous ground based procedures.

#### 4. INTEGRAL Reaction Wheel based New Safe Mode (NSM)

Once ESAM was disabled it was clear that a new safe mode was necessary to handle the spacecraft attitude in case of anomaly (e.g., RW switch off). The final objective is to patch the central computer (CDMU) with the NSM: an on-board based safe mode offers the advantage to show a fast reaction and would not be limited by the need of ground contact (INTEGRAL is based on real-time operations), solving the issue of commanding with potential intermittent telemetry link (as per ESAM9). Additionally, patching the CDMU and not the ACC makes NSM more reliable since it is completely independent of any AOCS (ACC) failure.

Two were the major challenges in CDMU NSM patch design:

1. the design and testing of the new patch, inclusive of NSM control algorithm and software logic;
2. the challenge to understand how to implement the patch in the on-board software contained in the CDMU.

The basic idea was to prototype the patch in JavaScript, language which has the advantage being compatible with the operational simulator. This allowed the flight control team to extensively test and modify the prototype design. Only after the finalisation of the patch, the JavaScript code was recorded into on-board software compatible language, ADA.

An overview of New Safe Mode concept is shown in Fig. 5. The Failure Detection Electronics monitors different parameters and assesses when a rate or an attitude anomaly is detected. In case a detection, the Automatic Reconfiguration Order (ARO) is asserted, up to this point there are no difference with old ESAM.

INTEGRAL on-board computer has some functionalities very similar with FDIR which were programmed to react to the ARO signal in such a way to command all instruments to a safe state, to power on the redundant ACC (ACC-B) and to automatically configure all units required for the New Safe Mode.

Once these preliminary actions are complete, the NSM first dumps all spacecraft rates and then uses telemetry from ACC to understand where the spacecraft is pointing and to take reaction wheels speeds and spacecraft rates. Thanks to these inputs, it computes the angular momentum that is necessary to point INTERGAL towards the Sun,  $h_{f,RW}$ . This value is passed to ACC, which commands the wheels accordingly until convergence is reached.

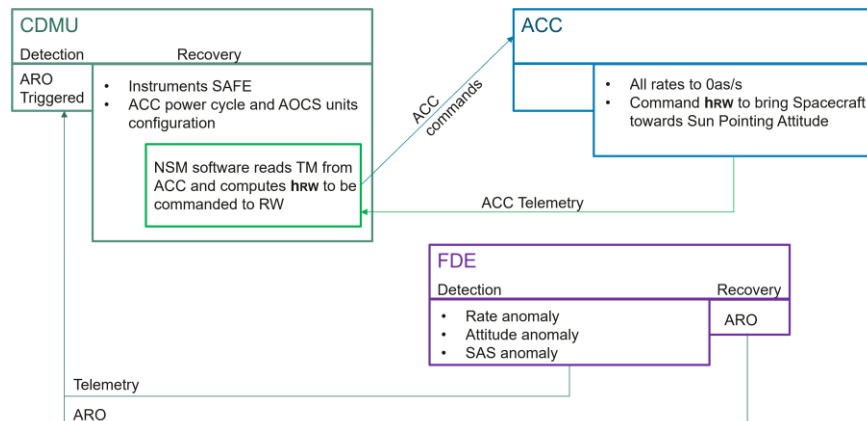


Fig. 5. CDMU New Safe Mode overview.

#### 4.1 On board software patch challenges and implementation

INTEGRAL was launched in 2002, twenty years ago, which means that its design was not compatible with the idea of the New Safe Mode. A feasibility study was carried out during the first phase of the NSM project, and three major challenges were identified:

1. Free on-board software memory resources were originally limited to 1.7Kword, which was estimated not to be enough to store the NSM patch;
2. INTEGRAL has limited pseudo-FDIR functionalities which allowed a maximum of three commands as reaction to certain events: these amount of TCs was not enough to configure all units needed by NSM;
3. Interfaces between CDMU and ACC needed to be enhanced in such a way to allow CDMU to read TM from ACC and to pass information to ACC how to command the Reaction Wheels;
4. NSM software logic and control algorithm needed to be robust.

This last point has been tackled developing a JavaScript prototype of the software patch that has been extensively tested on the operational simulator. Its detailed design is outlined in section 4.2.2, while the current section tackles the other three challenges.

The patch implementation was divided into 4 deliveries, each of them was addressing one of the points above.

##### 4.1.1 Spare memory area

As already introduced, industry estimated 1.7 KWords of free memory in the on-board software. It was also preliminarily estimated that a total amount of ~4KWords would be needed for NSM. The total CDMU memory area goes from 0 0000 to 3 FFFF hex. When the analysis was performed, only a portion of this area was mapped, from 0 0000 to 1 7FFF hex. Looking at the CDMU memory map it was found that there were another 27407 Words spare starting at physical address 1 8000 hex. The INTEGRAL CDMU uses expanded memory, with a Memory Management Unit providing alternative memory to the processor in different states. Address state 0 (in which all the CDMU software code runs) is currently not completely defined, with the potential to map 3 more pages (12KW) each of code and data space. It was therefore possible to re-map the Memory Management Unit to allow to address 24KW (from the 27407 words spare), Fig. 6.

In this memory area industry identified 6 pages of spare code, 3 pages at physical address 1 8000 hex, 1 9000 hex and 1 A000 hex that could potentially be used to store the NSM software. A first patch was provided to access and use this part of the memory and its accessibility and usability has been successfully tested using the operational simulator and the RF Suitcase.

		Physical			Logical (AS 0)	
		Start	End	Size (W, dec)	Start	End
PRAM	Code	0 0000	0 C942	51,523	0000i	C942i
	<i>Spare</i>					
	Constants	0 D000	0 EE85	7,814	1000o	2E85o
	<i>Spare</i>			379		
Data RAM	Ada Variables	0 F000	1 6493	29,844	3000o	A493o
	<i>Spare</i>			2,925		
	Heap	1 7000	1 7FFF	4,096	F000o	FFFFo
	<b>RAM Patch (PRAM)</b>	<b>1 8000</b>	<b>1 AFFF</b>	<b>12,288</b>	<b>D000i</b>	<b>FFFFi</b>
	<b>RAM Patch (Data RAM)</b>	<b>1 B000</b>	<b>1 DFFF</b>	<b>12,288</b>	<b>B000o</b>	<b>EFFFo</b>
	<i>Spare</i>			2,831		
	RTU_HKS (first NVB)	1 EB0E	1F3A6	2,200	-	-
	<i>Spare</i>			3,162		
	TIMETAG	2 0000	2 8983	58,000	-	-
	<i>Spare</i>			26,109		
	RTU_IRG_TAB	2 EF80	3 2058	12,500	-	-
	<i>Spare</i>			168		
	NPTMBUF	3 2100	3 B764	38,000	-	-
	<i>Spare</i>			156		
	PTMBUF	3 B800	3 BEE0	1,760	-	-
	<i>Spare</i>			16,672		

Fig. 6. Memory Map with extension.

##### 4.1.2 Extended pseudo-FDIR capabilities

INTEGRAL spacecraft has some sort of FDIR capabilities which are given by the combined use of an on-board monitoring function and a series of Re-Action Control Procedures, RACP. The spacecraft is endowed with an On-Board Monitoring (OBM) table which can contain up to 50 TM entries. Monitoring checks of all 50 TM are executed every 8 sec (CDMU cycle period) and in case any of these TM values is different from what is expected, a

corresponding Re-Action Control Procedure (RACP) is executed. In the original design, any RACP could contain up to 3 TCs, with a length of 125 words, for a maximum of 50TCs.

The objective was to use RACP to react on ARO signal to execute the following actions to reconfigure AOCS for NSM execution (only most relevant steps are reported):

- Both ACC powered OFF.
- Mode change from Inertial Pointing and Slew (IPS) AOCS mode used for routine operation to Stand-By Mode, where hardware reconfiguration will be performed.
- Redundant ACC powered ON (ACC-B), following the general FDIR principle to use redundant units to isolate failures. In case of non-convergence, a second RACP set is foreseen to use the nominal ACC (ACC-A), Fig. 7.
- All 4 Reaction wheel OFF
- Disable all Failure Detection (FDE) criteria.
- Power ON IMUs and Reactions Wheels. In nominal operations INTEGRAL uses only 3 of the 4 Reaction Wheels available, but NSM is design to use 4. Reconfiguration to three wheels is performed automatically the NSM software only in case one of them is failed.
- Power ON Fine Sun Sensors and Star Trackers used as prime, power OFF redundant Fine Sun Sensors and Star Trackers.
- Declare to ACC which are the units that needs to be used including IMU sets (redundant 1/3 for NSM RACP set 1 and 2/4 for NSM RACP 2 set) and Reaction Wheels.
- Mode change to Sun Sensor Acquisition (SSA) mode where it is possible to command directly Reaction Wheels with the desired angular momentum.
- Enable NSM.
- Configure NSM parameters (e.g., hardware to be used in the computation, maximum slew rate, convergence parameters, etc.).
- Trigger NSM.

These actions require a total of 73 TCs for the NSM RACP first set and 71 for NSM RACP redundant set (i.e., to configure ACC-A as prime, etc.). With the limitations given by the original design (3TCs for each on-board monitored TM and a maximum of 50TCs) this would not have been possible.

It was observed that, even if every TC was 125words long, they were mostly empty, therefore it was taken the decision to shorten each TC to 25words (considering headers), this allowed to 250 TCs in the same memory area. This was possible with the second patch release.

Additionally, each On-Board Monitor table entry can be linked to three TC pointers, therefore defining properly the NSM RACP TCs sets to be located at e.g., pointers from 100 to 172, it was possible to use two of the three pointers to define a range: in this way at ARO signal the On-Board Monitoring function would react sending TC every 2s from location 100 to 172.

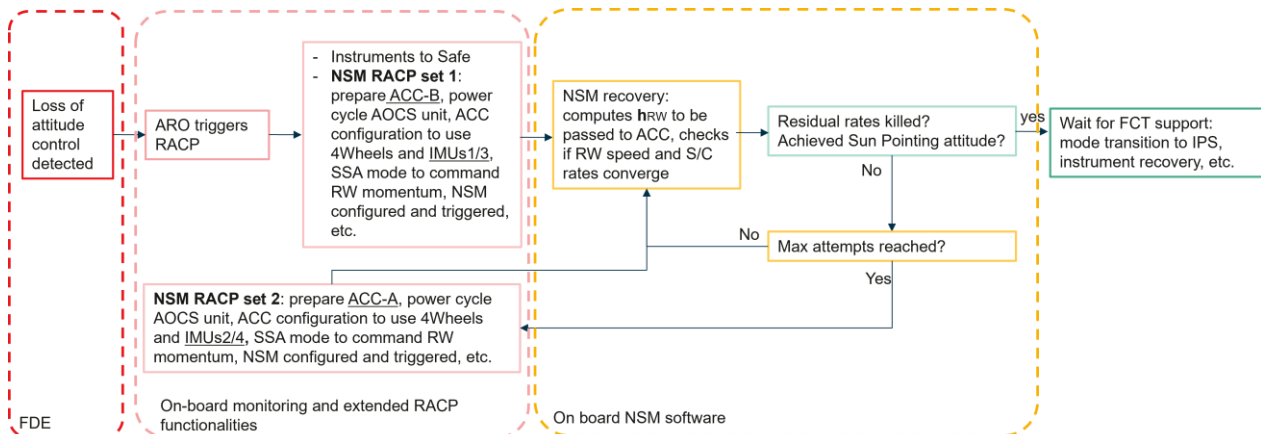


Fig. 7. RACP and NSM overview.

#### 4.1.3 CDMU/ACC interface

The interface patch delivery (called Input/Output delivery) aimed to verify:

- The possibility of writing into CDMU housekeeping packets NSM TMs

- The CDMU ability to read TM directly from ACC (fundamental for NSM control law) and to provide to ACC outputs (i.e., final angular momentum to be commanded from ACC to RW)

#### 4.2 NSM on-board software

The last step was the conversion of the JavaScript NSM controller software into ADA code (language used by the on-board software). A test campaign has been conducted to assure that the tests ran with the JavaScript prototype and the operational simulator were giving the same result as the resulted using the patch. All tests have been considered successful.

The NSM uses SASs to determine the relative position of the Sun with respect to the spacecraft. INTEGRAL is provided with three SASs:

- SAS1 is perpendicular to the +Z axis, which means that if Sun illuminates SAS1, then the solar arrays are illuminated (Fig. 1);
- SAS2 and SAS3 are perpendicular, respectively, to +Y and -Y axes (Fig. 1).

SASs outputs are expressed as  $\alpha$  and  $\beta$  (measured in currents, mA) which are respectively the current components measured along SAS Y ( $\alpha$ ) axes and SAS X axes ( $\beta$ ) respectively. The current along the two axis is generated by the Sun when it is illuminating the SAS surface: depending on how the Sun vector is oriented, the two currents components along the sensor X and Y axis change.

Spacecraft rates are measured by gyroscopes (IMUs) while RW command and speed measurement are provided by the RW driving electronics. These inputs are used by the control algorithm to command the RWs to change the spacecraft orientation toward the Sun with zero residual rates.

NSM baseline foresees to use all 4 RWs to stop the tumbling but this configuration is automatically changed if one of the units is identified as faulty or if ground control decides to use a different subset of RWs.

In case an anomaly is detected, like SEUs (affecting RW, Fine Sun Sensor or ACC), first the instruments are commanded to a safe state and then units are reconfigured for NSM (RACP, see section 4.1).

All hardware data are read (e.g., RWs speeds, IMU registered spacecraft rates, SASs outputs) and used by the control algorithm to determine which is the angular momentum to be applied to each RW to rotate the spacecraft towards a favourable Sun facing attitude. The basic equations used are reported below.

$$\mathbf{h}_{f,RW} = \mathbf{h}_{i,RW} + \mathbf{h}_{i,SC} - \mathbf{h}_{f,SC} \quad (1)$$

The Eq. (1) uses the momentum conservation to determine the final RW angular momentum ( $\mathbf{h}_{f,RW}$ ) to be commanded to the wheel to reach the desired attitude.  $\mathbf{h}_{f,RW}$  is expressed in spacecraft reference frame but the telecommand to force RW angular momentum needs this quantity in RW reference frame (i.e., inputs to the telecommand are one angular momentum value for each wheel).

Using the RW orientation matrix,  $\mathbf{A}$ , it is possible to obtain the RWs angular momentum values to be commanded to the RW to obtain the desired final spacecraft angular momentum ( $\mathbf{h}_{f,SC}$ ), needed to drive the spacecraft towards the correct attitude.

$$\mathbf{h}_{f,RW,RWframe} = \tilde{\mathbf{A}}^{-1} \mathbf{h}_{f,RW} \quad (2)$$

$\mathbf{A}$  is a 3x4 matrix which columns represents the orientation of each RW rotation axis with respect to spacecraft body frame. In case the recovery is executed using three RWs, then  $\tilde{\mathbf{A}}$  is equal to  $\mathbf{A}$  without the column corresponding to the RW that is not use (the result is a 3x3 matrix). In case four RW are used then  $\tilde{\mathbf{A}} = \mathbf{A}$  (3x4 matrix) and  $\tilde{\mathbf{A}}^{-1}$  is  $\mathbf{A}$  pseudo-inverse matrix.

Going back to Eq. (1), terms  $\mathbf{h}_{i,RW}$  (initial RW angular momentum) and  $\mathbf{h}_{i,SC}$  (initial spacecraft angular momentum) are determined, respectively, from RW speeds measurements and IMUs body rates readings, multiplying both terms by the respective inertias (i.e., RW and spacecraft inertias).

The final spacecraft momentum ( $\mathbf{h}_{f,SC}$ ) it is obtained multiplying the desired final spacecraft body rates ( $\mathbf{w}_f$ , provided by the NSM control law based on SAS readings) by the spacecraft inertia.

In the NSM current design, during the first cycles,  $\mathbf{w}_f$  is forced to 0 as/s while in all the subsequent cycles  $\mathbf{w}_f$  can be determined by means of using one of three possible control algorithms of increasing complexity, listed below. All use SASs readings to determine in which direction the spacecraft shall rotate.

1. Drift proposal: only one axes is controlled at the time, section 4.2.2.2.
2. Two-axis control: two-axis are controlled at the same time, section 4.2.2.

- Rescaling: this is used in case the two-axis control algorithm suggests commanding one RW beyond its maximal angular momentum capacity of +/-39Nms. In such a case  $w_f$  is reduced by a certain factor to avoid the RW saturation, section 4.2.2.1.

Due to time and budgetary constraints only option 2 (two-axis control algorithm) has been implemented in the on-board software, while option 1 (Drift Proposal) has been developed with ground automation. However all three options have been implemented and tested in JavaScript prototype.

The NSM comprises 7 different modes:

- 0 - Initialisation: initialise NSM configuration variables. The initial control law imposes to reduce rates to zero. NSM will stay in Init State unless "Triggered" to drive wheels.
- 1 - Drive Start: Calculate Momentum and send commands to drive the wheels based on the selected control law (i.e., Drift Proposal, 2-axis control or Rescaling)
- 2 - Driving: the wheels are commanded but not yet converged to expected wheel speeds (or not yet timed-out). Same applies to spacecraft rates. The current design foreseen to reach the Timeout after 15 cycles of non-convergence. Timeout will result in a Retry. In choosing the maximum value for timeouts it was considered the time needed by the reaction wheel to go from maximum angular momentum to minimum angular momentum (6min, with additional 2 minutes of margin, for a total of 8min).
- 3 - Drive End: when wheels speeds are met then spacecraft rates are checked against the expected ones. In case rates do not converge then the mode changes to Retry.
- 4 - Wait: all convergence criteria are met (i.e., wheels speed and spacecraft rates converge) and the algorithm waits a certain amount of seconds (currently set to 0) to go to Drive Start mode. This mode has been included to cover the case in which RW are commanded too often.
- 5 - Failed: if target momentum or rates not achieved after a certain number of reconfigurations (currently set to 1)
- 6 - Reconfigure: this mode is entered after the maximum number of maximum retries (currently set to 4). The software assumes that there are faulty/unresponsive units therefore it configures ACC to use different ones: it tries to recover attitude using NSM RACP set 2 (i.e., ACC-A, redundant IMU set). Only one reconfiguration is possible.
- 7 - Retry: retry driving the wheels with last control law output (do not select new one). Currently up to 4 retries are allowed before entering Reconfigure mode.

An overview of the modes is shown in Fig. 9. The control laws used will subject of section 4.2.2.

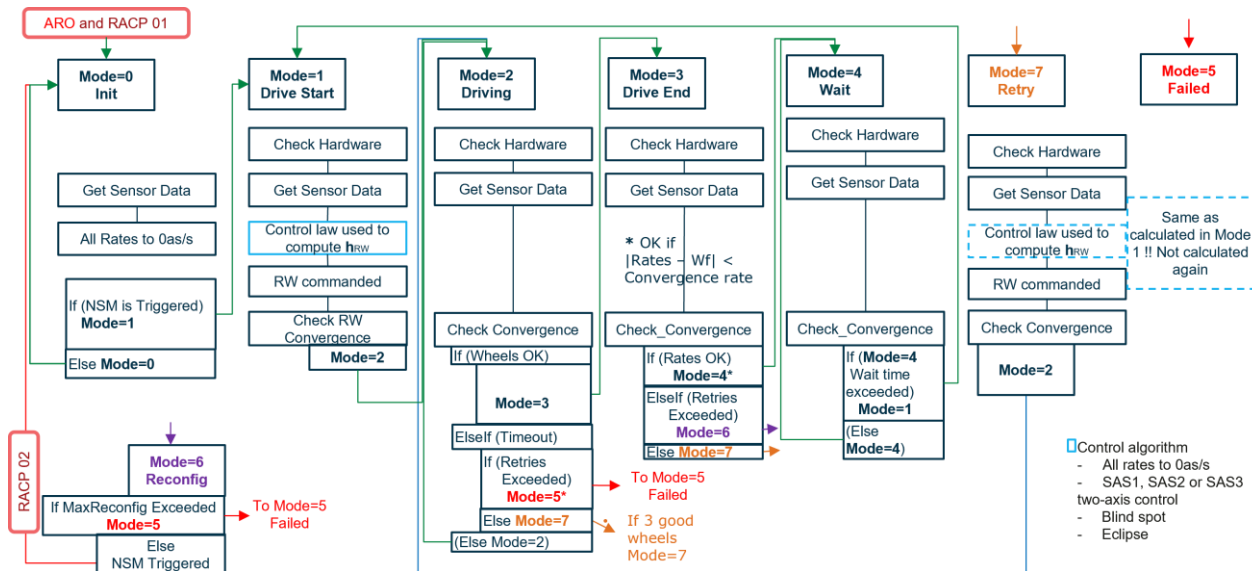


Fig. 8. NSM mode transition overview

It is foreseen to have one mode change for every CDMU cycle (8s), which means that in the best case the open loop duration of the algorithm (i.e., time interval between two Drive End mode) is 32s counting Drive Start, Driving, Drive End and Driving again. Hardware (IMU and RW) are checked at every cycle.

In the worst, it is 480s (8 minutes, multiply maximum timeout, by the maximum number of retries by 8s).

The final objective is to achieve the zero spacecraft rates to reach a stable Sun pointing attitude. Once this safe status is achieved, INTEGRAL recovery can continue from ground (e.g., payload recovery, ACC patches etc., IPS mode transition).

#### 4.2.2 NSM controller

The NSM controller computes the required RW target angular momentum to be passed to ACC only in Mode 1-Drive Start. In the current design, based on where Sun is present different algorithm are used (almost all two axis-controlled beside the Eclipse one). The Rescaling algorithm (section 4.2.2.1) has been implemented only in the JavaScript prototype but not in the on-board software patch due to monetary and schedule constraints. Drift proposal (Section 4.2.2.2) which is basically a single axis control approach, has been operational with ground automation since December 2021, as temporal substitute of the old ESAM which has been disabled after ESAM 9 (September 2021).

This paragraph will focus on the NSM controller version that has been implemented in the on-board software: the two-axis control approach (Fig. 9).

Sun is on SAS1 if:

- Sun present on SAS1 (Sun presence current lower threshold for all SASs is 10.4mA) or
- one of the SAS1 output currents is higher than a certain value  $I_D$  (configurable) and currently set to 5mA

Sun is on SAS2 or on SAS3 if:

- Sun is not on SAS1 and
- Sun present on SAS2/3 only or
- one of the SAS2/3 outputs currents is higher than a certain value  $I_D$  (configurable) and currently set to 5mA.

If Sun is not on any of the SAS, then the algorithm checks first for eclipse flag, and only in case it is confirmed that the spacecraft is not in Earth/Moon shadow, then it assumes that the condition is the blindspot.

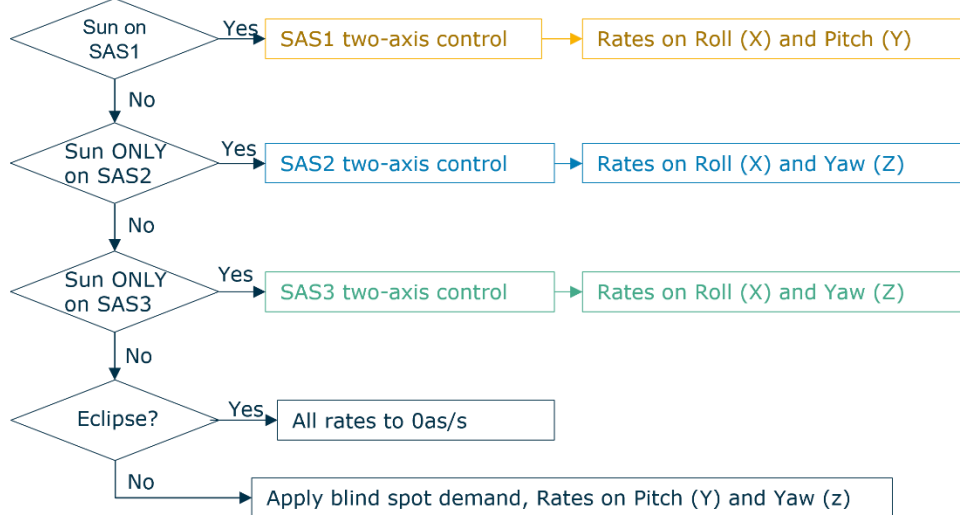


Fig. 9. Two axis control law overview.

The two-axis control algorithm consists in reading SASs output and using them to provide variable rates around two axis. Based on this, four different two-axis control laws are identified:

1. SAS1 two-axis control law, if Sun is visible on SAS1;
2. SAS2 two-axis control law, if Sun is visible only on SAS2;
3. SAS3 two-axis control law, if Sun is visible only on SAS3;
4. Blind spot control law.

In case the Sun is visible on SAS1 (no matter if it is also visible on SAS2/SAS3), the two-axis control is applied on roll and pitch. The control law reads the current output from SAS1 (in terms of  $\alpha$  and  $\beta$  angles expressed as a function of the current read from the SAS, [mA]) and brings the Sun in the centre of SAS1 field of view. The rotation rate around the two axis is progressively reduced once the Sun gets close enough to the SAS1 FoV centre (i.e.,  $\alpha = 0\text{mA}$  and  $\beta = 0\text{mA}$ ): currently the rotation rate starts decreasing when SAS1 readings are lower than 10mA (configurable, expressed as  $i_{threshold}$  in the equations). Rates decreases proportionally to  $\alpha/\beta$  readings. Maximum rotation

rates around both axis is 200 as/s (configurable, expressed as  $w_{max}$  in the equations). The value 200 as/s is the one used to execute slews during routine operations.

Fig. 10 shows an example of the SAS1 two-axis control algorithm. Yellow area indicates SAS1 FoV, while blue and green areas indicate, respectively, SAS2 and SAS3 FoVs (the same colour logic is applied in Fig. 11 and Fig.12). When the Sun is present on more than one SAS, it is located in areas where two colours superpose.

$w_f$  components roll and pitch are computed by the following equations, while the yaw component is  $w_z = 0$ :

$$w_x = -sign(\alpha) * \left( \frac{\min(i_{threshold}, |\alpha|)}{i_{threshold}} \right) * w_{max} \quad (3)$$

$$w_y = -sign(\beta) * \left( \frac{\min(i_{threshold}, |\beta|)}{i_{threshold}} \right) * w_{max} \quad (4)$$

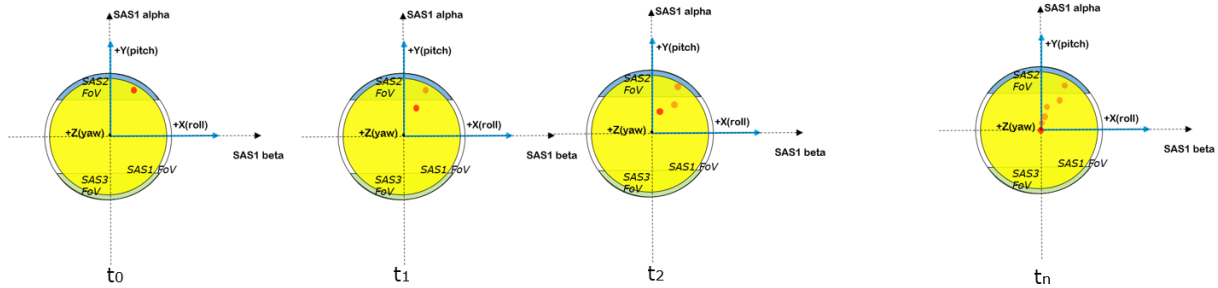


Fig. 10. Example of SAS1 two-axis control law. At the beginning ( $t_0$ ) Sun is found on both SAS1 and SAS2 (up-rightmost light red dot). Each light red dot represents a subsequent Sun position on the SAS. The algorithm drives the Sun towards the centre of SAS1 (dark red dot) with decreasing rates (approaching the centre the dots are closer to each other,  $t_n$ ).

When Sun is visible only on SAS2, the two-axis control algorithm brings the Sun towards the area of SAS2 FoV, where Sun would also be visible on SAS1 thanks to the large field of view of the sensors. At this point the SAS1 two-axis control law would take over the spacecraft attitude control. The SAS2 two-axis control law applies rates on roll and yaw bringing the  $\beta$  angle reading to 0mA (yaw rotation) and the  $\alpha$  reading towards SAS2 area where Sun is also visible on SAS1 (Fig. 11 area where blue and yellow superpose). Also, in this case the rotation rate around yaw decreases proportionally with SAS2  $\beta$  readings when less than  $i_{threshold}$  (10mA, i.e., Sun is getting closer to SAS2  $\beta = 0$ mA). Maximum rotation rates around both axis is 200 as/s ( $w_{max}$ , configurable).

$w_f$  components roll and yaw are computed by the following equations, while the pitch component is  $w_y = 0$ :

$$w_x = -w_{max} \quad (5)$$

$$w_z = sign(\beta) * \left( \frac{\min(i_{threshold}, |\beta|)}{i_{threshold}} \right) * w_{max} \quad (6)$$

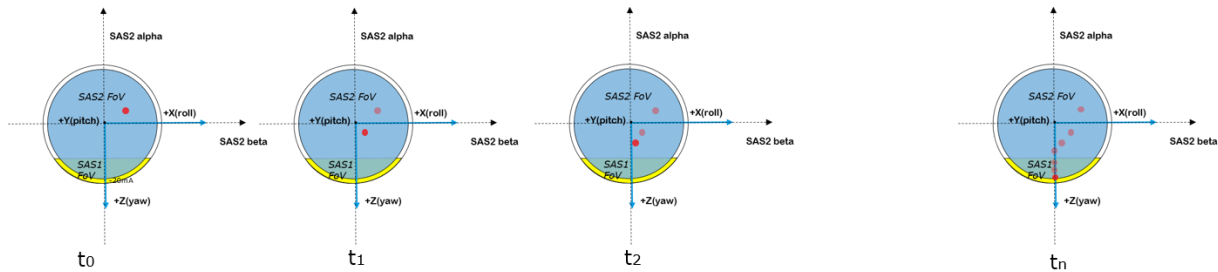


Fig. 11. SAS2 two-axis control representation.

Similarly, when the Sun is visible only on SAS3 (Fig. 12), the two-axis control law brings the Sun towards the area of SAS3 FoV where Sun would also be visible on SAS1 (target is to bring  $\beta$  reading to zero and modify  $\alpha$  angle reading towards SAS3/SAS1 overlap region). Also in this case, a constant rate on roll is applied, while the yaw rate is progressively reduced as the Sun is approaching the area where it is visible also on SAS1. Maximum rates are 200 as/s ( $w_{max}$ , configurable).  $w_f$  roll and yaw components are computed by the following equations, while the pitch component is  $w_y = 0$ :

$$w_x = w_{max} \quad (7)$$

$$w_z = -sign(\beta) * \left( \frac{\min(i_{lim}, |\beta|)}{i_{lim}} \right) * w_{max} \quad (8)$$

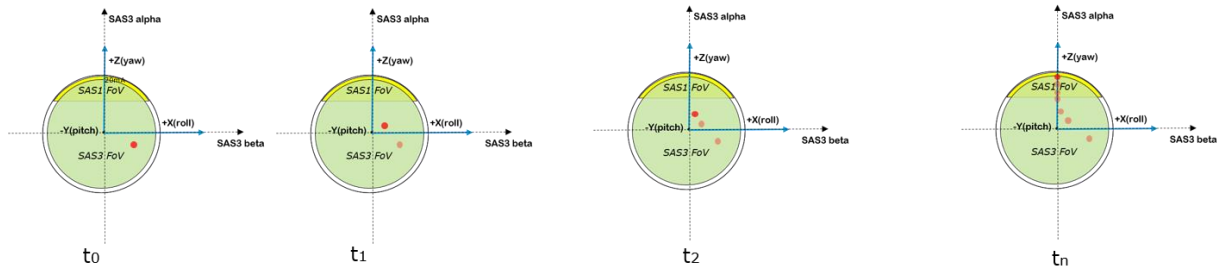


Fig. 12. SAS3 two-axis control representation.

In case of a blind spot (Sun non-present in any of the SASs but no Eclipse, Fig. 12), fixed rates are applied:

$$w_f = \left[ 0, -w_{max}, -\frac{w_{max}}{2} \right] \quad (9)$$

This approach is based on the old ESAM controller for which pitch rate is -720 as/s and yaw rate is -360 as/s. Since the NSM is fully based on RW, the rates used by ESAM were not applicable therefore a tailored version of ESAM controller for blind spot has been implemented. Tests demonstrated that there are few blind spot cases in which applying one axis control law result in a ping pong effect for which the Sun cannot be acquired stably on one SAS. This issue is solved using a two-axis control algorithm.

To justify some of the default parameters introduced in this section, the SAS model needs to be further detailed. At the beginning of life the maximum SAS output current value was around 33mA ( $i_{max}$ ). Analysis on the ESAM9 made clear that currently, due to radiation degradation,  $i_{max} = 19\text{mA}$ . It is expected a further  $i_{max}$  degradation until 2029 (INTEGRAL re-entry). The SAS model is represented by the equations below between  $0^\circ$  and  $83^\circ$ , from  $83^\circ$  to  $97^\circ$  the output decreases linearly from  $I_{max}$  at  $83^\circ$  to  $0\text{mA}$  at  $97^\circ$ .

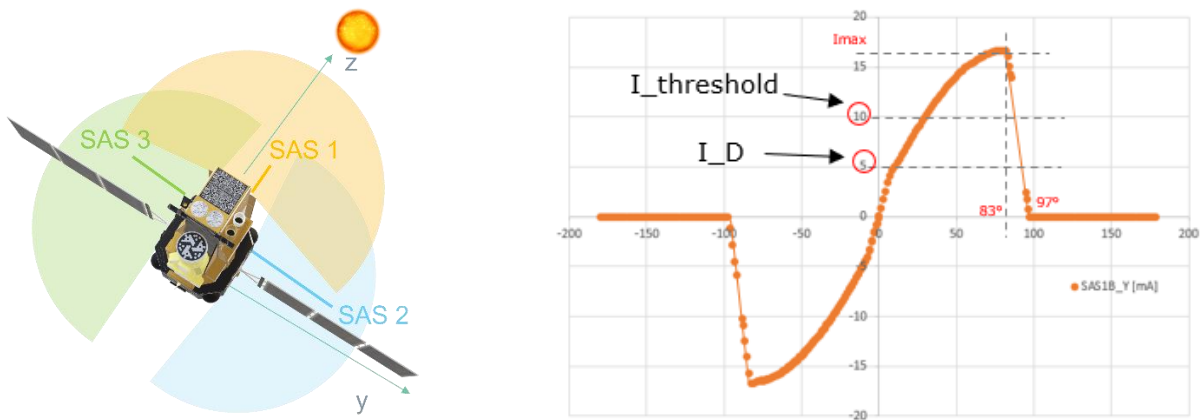


Fig. 13. Representation of complete SAS model: on the left the SASs Field Of Views, on the right it is represented SAS1 beta current against beta angle.

If  $i_{max} < i_{threshold}$  then rates would never be equal to  $w_{max}$  because the SAS current value for which rates start decreasing is always reached (e.g., in SAS1  $w_x$  control law the  $\min(i_{threshold}, |\alpha|)$  would always be  $|\alpha|$ ).

If  $i_{max} < i_D$  then only the Sun presence information would be used to determine on which SAS the Sun is located. Since the Sun presence information cannot be changed (10,4mA), with further degradation of the SAS the result would be creation of blind spot between SAS1 and e.g., SAS2. The value  $i_D=5mA$  was chosen to have this constraint respected also with SAS max degradation at end of life (2029). However, setting  $i_D=5mA$  could potentially lead to the situation in which roll rates decreases when there is the passage between the SAS2/3 two-axis control and SAS1 two axis control. This case has been simulated and analysed and it is considered unlikely to have angles above  $83^\circ$ , case for which this situation is happening since nominal attitudes never goes beyond  $\pm 5^\circ$  on roll and  $\pm 40^\circ$  on pitch.

Once the recovery is finished, when the spacecraft FSS readings on both axis are within the  $\pm 5^\circ$  for a certain amount of cycle, there is the option to command a fixed yaw rate (currently 10as/s) to counteract RW momentum drifting caused by solar radiation pressure while waiting for the flight control team to intervene to perform recovery from NSM, i.e., Inertial Pointing and Slew ACC mode transition, instruments recovery, etc.

#### 4.2.2.1 Rescaling

The "rescaling" feature was designed to update the RW angular momentum if the demanded  $w_f$  is higher than the RW capacity as this scenario would lead to saturation of one of the RWs. This is achieved fixing the highest RW angular momentum to  $\pm 37Nms$  (i.e., the maximum momentum) and re-scale down all others in such a way to keep  $h_{f,SC}$  vector direction but modifying its magnitude (Fig. 14). Currently rescaling is not part of the on-board software implementation but it has been considered in the JavaScript prototype and it is not excluded that in future the software will be modified with this feature. Tests have shown that limiting the plannable angular momentum (e.g., in routine operation the total of 15 Nms is never exceeded), during the recovery the value of 37Nms was never reached. Realistically speaking it would be very unlikely to hit conditions for which rescaling would be needed.

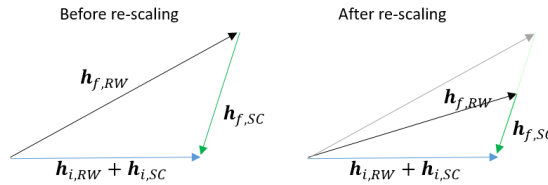


Fig. 14. Simplified 2D representation of rescaling.  $Scalefactor = 0.5$ . Reference equation is Eq. (1).

The result is a spacecraft rotating in the right direction (towards the Sun) but with different rates than the ones computed by the two-axis controller. In other words vector  $h_{f,SC}$  is multiplied by a constant called  $Scalefactor$  and indicated in the equation as  $S$ . Starting from Eq. (1), the  $Scalefactor$  value is computed in the following way. The quantity  $h_{temp}$  is defined as:

$$h_{temp} = h_{i,RW} + h_{i,SC} \quad (10)$$

so that Eq. (1) can be re-written as:

$$h_{f,RW} = h_{temp} - h_{f,SC} \quad (11)$$

The  $Scalefactor$  is applied to  $h_{f,SC}$  in such a way to maintain  $h_{f,SC}$  direction, changing only its magnitude:

$$h_{f,RW} = h_{temp} - S * h_{f,SC} \quad (12)$$

Eq. (12) is expressed in spacecraft reference frame but to compute  $Scalefactor$  this equation needs to be expressed in RW reference frame using Eq.(2)., which leads to:

$$h_{f,RW,RW \text{ frame}} = \tilde{A}^{-1} h_{temp} - S * \tilde{A}^{-1} h_{f,SC} \quad (13)$$

Considering  $n$  being the RW with highest momentum that exceeds  $39Nms$ , value identified as  $h_{max}$  ( $h_{max} = \text{sign}(h_{f,RW,RW \text{ frame}}(n)) * 39Nms$ ),  $n$  component of Eq. (13) can be written as:

$$h_{f,RW,RW \text{ frame}}(n) = \tilde{A}^{-1}(n,:) h_{temp} - S * \tilde{A}^{-1}(n,:) h_{f,SC} \quad (14)$$

where  $\tilde{\mathbf{A}}^{-1}(n, :)$  is a 1x3 vector and  $\mathbf{h}$  is 3x1 vector. Multiplying  $\tilde{\mathbf{A}}^{-1}(n, :)$  by  $\mathbf{h}$  is equivalent with performing the scalar product between these two vectors, therefore Eq. (14) is a scalar equation with  $S$  unknown. Therefore,  $S$  value can be retrieved as:

$$S = \frac{\tilde{\mathbf{A}}^{-1}(n, :)\mathbf{h}_{temp} - h_{max}}{\tilde{\mathbf{A}}^{-1}(n, :)\mathbf{h}_{f,SC}} \quad (15)$$

In case any of the wheels is still saturated after rescaling is applied or if  $S < 0$  or  $S > 10$ , then the angular momentum of that RW is forced to  $h_{max}$  and rescaling is not applied. This approach is needed to avoid the spacecraft to rotate in the wrong/opposite direction ( $S < 0$ ) or to prevent the algorithm to diverge, causing instability ( $S > 10$ , value configurable).

In the *Scalefactor*  $S$  equation (Eq. 15), the denominator can be equal to zero also for some specific combination of RW failure and saturation, therefore it needs to be applied carefully. Inserting in the algorithm a term that can give an infinite result, like in this case, would cause an overflow that would result in CDMU halt.

#### 4.2.2.2 Drift proposal control algorithm

The Drift Proposal control algorithm is also part of the JavaScript prototype, but it has been descoped from the on-board software implementation in early stage of the project. However, this is the control law currently implemented in the ground automated safe mode, operational since December 2021.

The drift proposal control algorithm consists in identifying on which SAS the Sun is present and, based on this, it provides a fixed rate around one of the axes. Only in case of blind spot (i.e., Sun in none of the SASes) the algorithm provides a fixed rotation around two axes.

Below are listed the drift proposals used by the algorithm ( $\mathbf{w}_f$  expressed in as/s).

1. If Sun is on SAS1 and SAS2, then  $\mathbf{w}_f = [-400, 0, 0]$
2. If Sun is on SAS1 and SAS3, then  $\mathbf{w}_f = [+400, 0, 0]$
3. If Sun is on SAS1 only and roll and pitch angles are into +/- 5°, then  $\mathbf{w}_f = [0, 0, 0]$
4. If Sun is on SAS2 only, then  $\mathbf{w}_f = [-400, 0, 0]$
5. If Sun is on SAS3 only, then  $\mathbf{w}_f = [+400, 0, 0]$
6. If eclipse flag is high, then  $\mathbf{w}_f = [0, 0, 0]$ . This drift proposal is kept until the eclipse is finished.
7. If Sun is not present in any of the SASs and eclipse flag is low, it means that Sun is in the blind spot, then  $\mathbf{w}_f = [0, -400, -200]$
8. If Sun is on SAS1 only and roll angle is close to +5°, then  $\mathbf{w}_f = [-200, 0, 0]$  to bring roll angle into +/- 5°
9. If Sun is on SAS1 only and roll angle is close to -5°, then  $\mathbf{w}_f = [+200, 0, 0]$  to bring roll angle into +/- 5°
10. If Sun is on SAS1 only and pitch angle is close to +5°, then  $\mathbf{w}_f = [0, -200, 0]$  to bring pitch angle into +/- 5°
11. If Sun is on SAS1 only and pitch angle is close to -5°, then  $\mathbf{w}_f = [0, -200, 0]$  to bring pitch angle into +/- 5°

#### 4.3 Testing

In order to first test the NSM JavaScript prototype and later the patch with the operational simulator, some test cases have been identified as combination of worst initial attitude and worst RW angular momentum distribution.

Regarding RW momentum, three different cases have been considered, as reported hereafter.

1. Total of 15.3Nms which is the total plannable angular during nominal operations. This total momentum is divided along spacecraft axis in such a way that  $h_x$  and  $h_y$  are less than |10|Nms and  $h_z < |6|$ Nms. Realistically, for nominal operations, single RW speed does not exceed |2000|rpm (wheel saturation is reached at |4000|rpm). This constraint is considered in choosing a subset of different initial RW speed distribution giving a total angular momentum of 15.3Nms.
2. Total of 22Nms which realistically represents the maximum total angular momentum plannable in case a specific (unplanned) target of sudden interest needs to be observed. In this case  $h_x$  and  $h_y$  are less than |15|Nms and  $h_z < |6|$ Nms. In choosing the initial RW speed distribution, the single RW speed cannot exceed |2500|Nms.
3. Total of 27Nms as worst-case scenario. In this case  $h_x$  and  $h_y$  are less than |19|Nms and  $h_z < |6|$ Nms. In choosing the initial RW speed distribution, the single RW speed cannot exceed |3000|Nms.

In choosing the RW speed distribution set giving 15.3Nms, 22Nms and 27 Nms, only RW2, 3 and 4 have been considered (RW1 off, this is the set used in nominal operations), Fig. 15.

Initial pointing considered for testing are given by attitudes slightly outside from the operational range (grey area in Fig. 15), but within the Fine Attitude Anomaly detector boundaries, to avoid ARO triggering before any failure is injected (i.e., an attitude beyond FAAD leads to ARO).

extreme realistic attitudes. Because of the instrument thermal stability and the geometry of the solar arrays, the spacecraft is constraint to operate with a roll angle of  $\pm 5^\circ$  and a pitch angle of  $\pm 40^\circ$ . One of the sensors used by the FDE (the Fine Attitude Anomaly Detector) triggers the ARO signal for FSS angles equal to  $FSS_\alpha = |12|^\circ$  and  $FSS_\beta = |45|^\circ$ . Based on this, the chosen worst case starting attitudes are  $FSS_\alpha = \pm 10^\circ$  and  $FSS_\beta = \pm 40^\circ$ : the aim is to have extreme attitude without triggering ARO. These values correspond to an initial SAS1 reading of:  $SAS1_\alpha = \pm 8\text{mA}$  and  $SAS1_\beta = \pm 24.8\text{mA}$ .

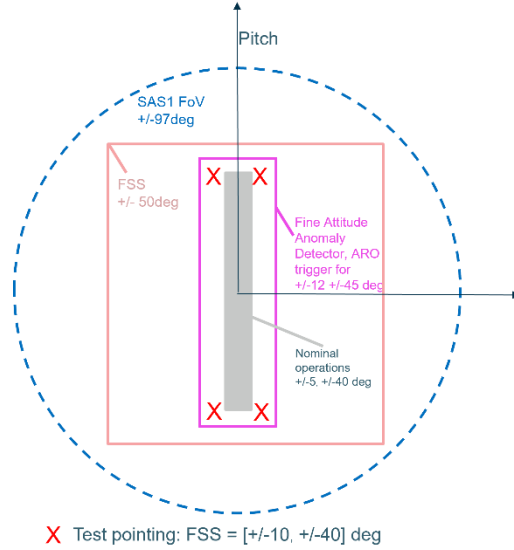


Fig. 15. Example of momentum distribution in case of targets of sudden interest (total angular momentum of 22Nms) on the left. On the right test pointing justification.

To obtain these extreme attitudes, a modified version of the NSM control law has been used: given target  $SAS1_\alpha$  and  $SAS1_\beta$  the control law commands the RWs speed to achieve the desired orientation. In Eq. (16) and Eq. (17)  $\alpha_t$  and  $\beta_t$  represents the target SAS1  $\alpha$  and  $\beta$  readings (e.g.,  $-8\text{mA}/-24.8\text{mA}$ ). Terms  $\alpha_{err}$  and  $\beta_{err}$  will be used to determine when the algorithm should start decreasing the spacecraft rate when  $\alpha$  and  $\beta$  are approaching the desired  $\alpha_t$  and  $\beta_t$  values. The algorithm stops when  $\alpha_t = \alpha \pm 1^\circ$  and  $\beta_t = \beta \pm 1^\circ$ .

$$\alpha_{err} = \alpha_t - \alpha \quad (16)$$

$$\beta_{err} = \beta_t - \beta \quad (17)$$

In case  $|\alpha_{err}| > I_{lim}$  and  $|\beta_{err}| > I_{lim}$  where in this case  $I_{lim} = 5\text{mA}$ , then the control laws look respectively as it is shown in Eq. (18) and (19),

$$w_x = \text{sign}(\alpha_{err}) * w_{max} \quad (18)$$

$$w_y = \text{sign}(\beta_{err}) * w_{max} \quad (19)$$

otherwise, the equations applied are (respectively) Eq. (20) and (21).

$$w_x = \text{sign}(\alpha_{err}) * \left( \frac{\alpha_{err}}{i_{lim} * f} \right) * w_{max} \quad (20)$$

$$w_y = \text{sign}(\beta_{err}) * \left( \frac{\beta_{err}}{i_{lim} * f} \right) * w_{max} \quad (21)$$

The  $f$  and  $i_{lim}$  in Eq. (20) and (21) comes from the trade-off between reaching rapidly the initial attitude for the test cases avoiding overshooting. The aim is to start decreasing the rates when the readings on SAS1 are  $i_{lim}$  mA from the desired ones (i.e.,  $5\text{mA}, 10\text{mA}/20\text{mA}$  for NSM). The maximum rate is kept for longer with respect to the NSM case

in order to reach the final attitude fast enough. Tests demonstrated the necessity of breaking these rates faster than NSM, to avoid overshooting with consequences like ARO being triggered. For this reason, after multiple simulations, it was added a factor  $f = 5$ .

In the tests also the impact of solar radiation pressure is included, to cover cases in which the spacecraft is left drifting with no control for a certain amount of time. For this purpose, a constant, representative, a solar radiation pressure torque in the order of magnitude of  $10^{-6}$  Nm on roll and  $10^{-5}$  Nm on pitch and yaw is applied. In the tests also realistic IMU drift and noise are considered as well as eclipse cases.

So far only realistic scenarios have been considered, however, to assess NSM robustness also blind spot pointing have been created starting from a perfectly Sun pointing spacecraft and forcing  $-/+90^\circ$  rotation around pitch and then  $-180^\circ$  rotation around roll. With these three test cases it is possible to observe the blind spot algorithm applied followed by, respectively, SAS2+SAS1, SAS1 and SAS3+SAS1 control laws respectively.

A wide variety of failures were injected in the simulator to assess the ability of the NSM to regain attitude control. Amongst them:

- RW SEU failures
- RW permanent failures
- IMU SEU failures
- IMU permanent failures
- ACC OFF
- Spacecraft power unit reset

Test plan foresees also double failure cases (e.g., RW and following IMU permanent failures).

Simulator tests have been fundamental to verify NSM functionalities, debug the algorithm logic, for fine tuning and to prove that the delivered software behaves like the JavaScript prototype.

The example shown below in section 4.3.1, is a NSM recovery upon detection of a RW3 spurious switch off (a typical signature of SEU). This simulation proves that NSM would have been able to recover, an event like the one that caused ESAM9 (RW3 SEU) within 30 minutes. Drift Proposal (DP) are used in the NSM on board software only as an indicator of Sun location:

- DP = 3: if Mode = 0 and rates are commanded to 0as/s or if Sun is  $+/-5^\circ$  on FSS on both axis (Sun close to SAS1 field of view, yellow area in Fig. 16). In this second case SAS1 two-axis control is applied.
- DP = 4: if Sun is only on SAS2, SAS2 two-axis control is applied
- DP = 5: if Sun is only on SAS3, SAS3 two-axis control is applied
- DP = 6: if spacecraft is in eclipse, all rates are commanded to 0as/s
- DP = 7: Sun is in the blind spot and blind spot control law is applied
- DP = 8: if Sun on SAS1 but above  $+/-5^\circ$  on FSS. SAS1 two-axis control is applied.

For control algorithm, please refer to section 4.2.2.

#### 4.3.1 ESAM9-like recovery from RW3 SEU

Fig. 16 to Fig. 19 show how an ESAM9-like case would have been recovered by the on board NSM software. Dots in Fig. 16 represent Sun position on sensors (i.e., SASs).

The test presents worse initial condition than the ones before ESAM9 in September 2021:

- $FSS_\alpha = -10^\circ$  and  $FSS_\beta = -40^\circ$  (test) against  $-2.06^\circ$  and  $FSS_\beta = -16.52^\circ$  (pre-ESAM9).
- RW speed distribution is  $[0, +990, +1414, -1839]$ rpm with a total angular momentum of 15.3Nms (test) against  $[0, +458, +927, +220]$ rpm with a total angular momentum of 10.77Nms (pre-ESAM9).

In Fig. 16, the red dot shows the initial position of the Sun, which is an indicator of the pointing of the spacecraft, which is outside the nominal operations area (in grey) but inside the Fine Attitude Anomaly Detector used to trigger ARO. At the beginning of the simulation, an RW3 OFF is simulated (SEU) and the spacecraft starts drifting (Fig. 17, rates chart prior NSM Enable) as the spacecraft absorbs the angular momentum previously stored in RW3. It takes almost 500s for the FDE to trigger an ARO based on Fine Attitude Anomaly Detector (FSPAAD). In this period, we can see the satellite drifting around roll (light blue dots between red and yellow ones, Fig. 16). Please consider that before NSM is triggered DP is set as default to 3 but reaction wheels are not commanded with 0as/s final rate target (Fig. 17, Drift Proposal chart). RACP 1 set is executed before NSM is triggered. Once this happens, there is a brief period where all rates are commanded to 0 (Mode = 0 Init and DP = 3)

Thereafter, there is a Mode transition to 1 (Drive Start) and SAS1 control law starts to be applied (blue dots, DP = 8) bring with a direct path Sun towards SAS1 FoV centre. Roll rate (Fig. 17) is decreasing since the mode transition (meaning that SAS1  $\alpha$  is less than 10mA, Fig. 18), while pitch rate is initially briefly equal to the maximum (currently 200as/s) because SAS1  $\beta$  is higher (Fig. 18) and then decreases proportionally with  $\beta$ . When both FSS angles goes

below  $5^\circ$  then there is a further DP change to 3 (Fig. 17, spacecraft angle and Drift Proposal charts): this does not translate in a change of control law but it is only used the software to increase the Sun Safe Counter at every cycle. When this counter is above a certain value (configurable, set to 40 in the test), then the fixed yaw rate can be applied counteract RW momentum drift induced by solar radiation pressure (Fig. 17, rates chart, right-most part).

Long-duration tests have been executed and they show that the spacecraft can stay in this state for hours, waiting for the flight control team to perform the recovery post NSM (i.e., AOCS mode change to IPS and instrument recovery).

Good indicators to assess if the recovery was smooth are the modes used by NSM and how directly Sun is brought towards SAS1 centre. In this specific test case, we can observe in Fig. 16 that the Sun path is very linear and that the modes used don't go above 4 (Fig. 19), which represent a very nominal and smooth case.

Full recovery is performed in 30 minutes from NSM triggering and Sun pointing is kept with high precision. ESAM9 recovery took many hours and old ESAM concept was coarser than NSM.

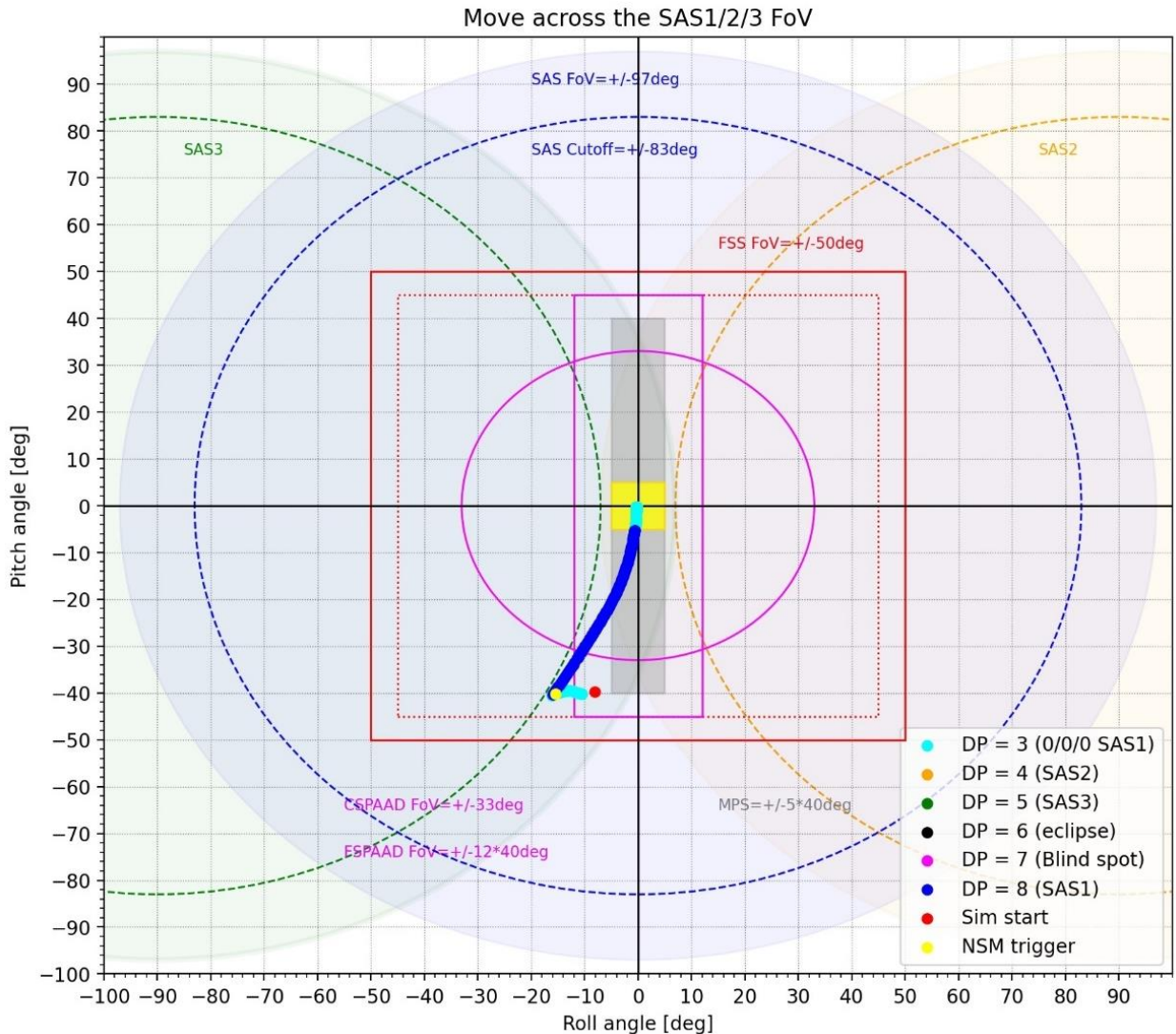


Fig. 16. Move across SAS1/2/3 field of view in case of ESAM9-like RW3 SEU.

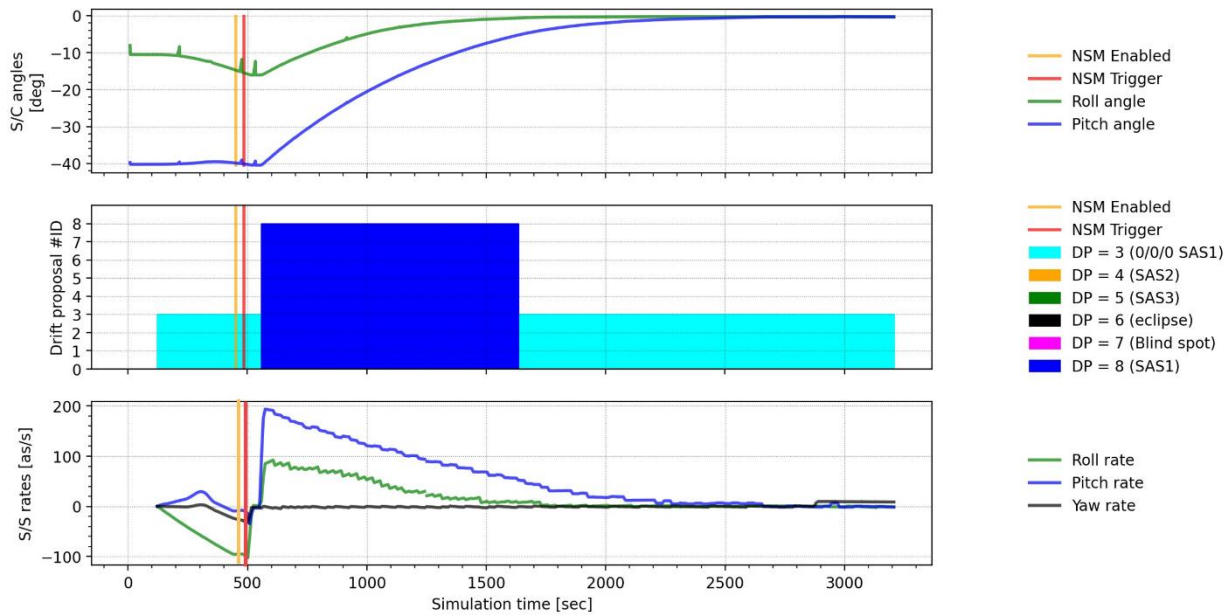


Fig. 17. Spacecraft angles, Drift Proposal and spacecraft rates for ESAM9-like RW3 SEU test case.

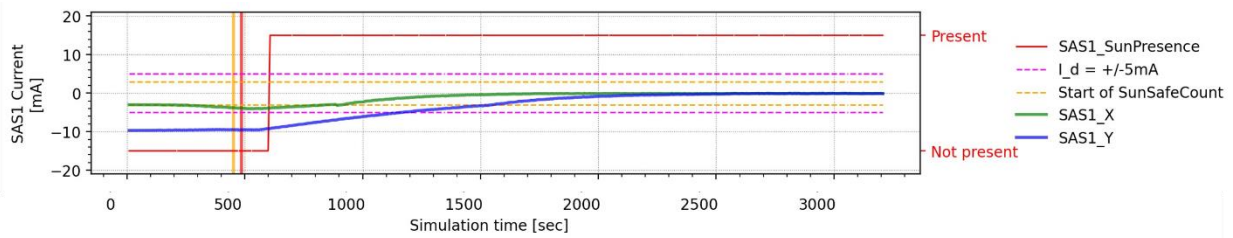


Fig. 18. SAS1 current for ESAM9-like RW3 SEU test case. SAS1\_X represents SAS1  $\alpha$ , while SAS1\_Y represents SAS1  $\beta$ .

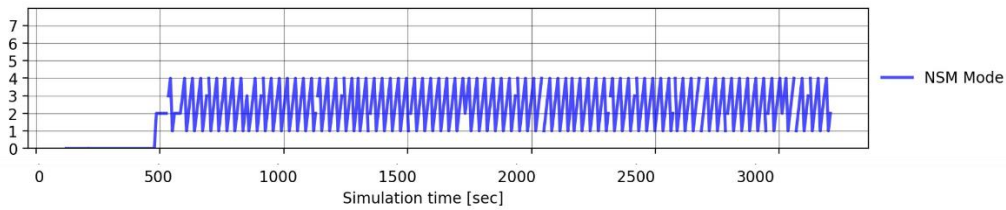


Fig. 19. Modes change for ESAM9-like RW3 SEU test case.

#### 4.3.2 Eclipse test case with RW3 SEU

In this section an eclipse test case example is shown. Initial pointing is given by  $FSS_{\alpha} = -10^{\circ}$  and  $FSS_{\beta} = -40^{\circ}$  and RW speed distribution is  $[0, +990, +1414, -1839]$ rpm with a total angular momentum of 15.3Nms (test) against  $[0, +424, +2687, +424]$ rpm with a total angular momentum of 27Nms (worst case scenario).

The eclipse starts at the beginning of the test, like the previous test example DP is 3 before NSM triggers because this is the default value, but wheels are not commanded, and spacecraft rates are not going to 0as/s.

At the beginning of the test RW3 is switched OFF (SEU). Spacecraft rates increases (as per previous case) but being in eclipse the FDE cannot use Coarse Anomaly Attitude Detectors which work with the Sun, therefore in this case ARO is issued by the FDE only when one of the rates reaches 360 as/s (Fig. 20, pitch rate). Once NSM is triggered rates are duped to 0as/s while DP = 3, Fig. 21 shows that one retry is needed (i.e., same control law is applied twice).

Afterwards the algorithm select DP = 6, Eclipse case, because no SASs see the Sun and Eclipse flag is present: rates are kept to 0as/s for multiple cycles. Only after eclipse exit SAS1 two-axis control law can be used (DP = 8 and DP = 3). Recovery takes 30 minutes.

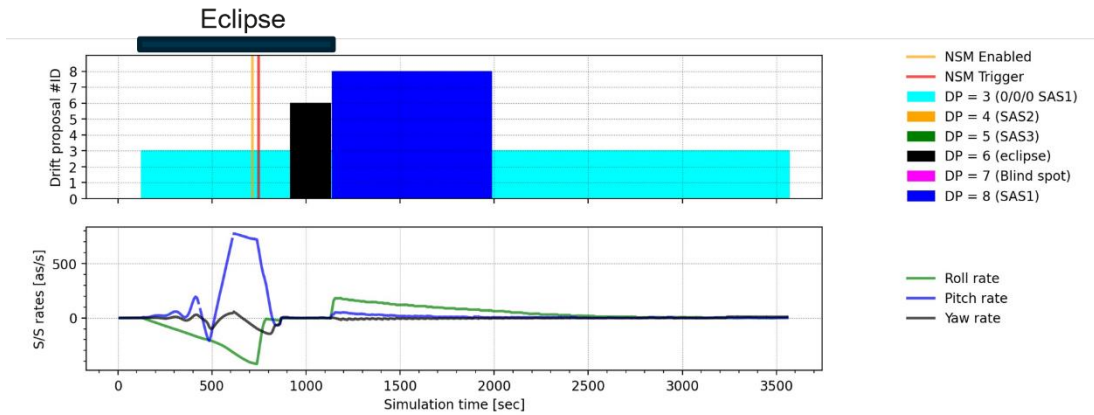


Fig. 20. Spacecraft angles and Drift Proposal for eclipse test case.

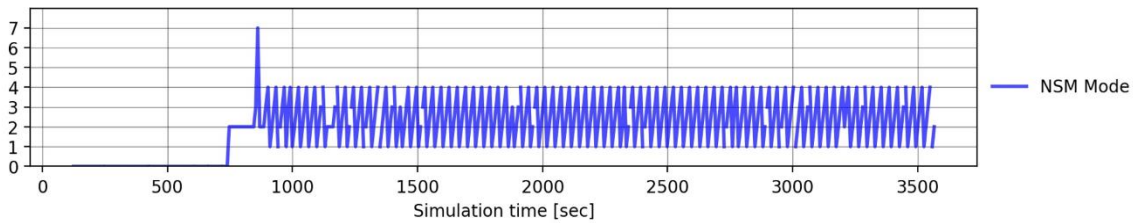


Fig. 21. NSM Modes change for eclipse test case.

#### 4.3.3 Permanent failed unit test case

This section shows a test where RW3 is permanently failed. The failure is simulated as a bus error (i.e., error on the communication line between ACC and on-board computer). Initial pointing and RW angular momentum are the same as the one used in section 4.3.2 (Eclipse case).

When NSM is triggered, red dot in Fig. 21, first the software tries to bring rates to 0as/s using all 4 reaction wheels. In the Retries chart it is possible to see that the wheel speed and spacecraft rates achieved don't match the desired ones. The Number of Retries increases (Mode = 7) until the software understands that it needs to exclude RW3 from the computation, and it changes set from 1234 (4 wheels) to 124. At every cycle the NSM software checks the hardware and in case one wheel is not powered up it will change immediately the wheel set without running RACP set 2. In this test case the failure is not simulated with RW3 unpowered but with an error on the communication bus. Therefore the software waits up to 4 retries before changing the wheel set to exclude RW3 to make sure that the bus error is an actual failure.

This initial non-convergence scenario is reflected in the movement across the SASs (Fig. 22) where we observe that, while in DP = 3 (rate to 0as/s) the path of the Sun is not directed straight towards SAS1 centre. Once the RW set is changed, then convergence is immediately reached, and rates goes immediately to 0as/s. DP is changed to 8 and SAS1 two-axis control is applied bringing the Sun with a straight path to the SAS1 centre. Recovery takes around 40minutes.

In this case NSM RACP 2 set was not executed (Mode never changed to 6-Reconfigure) because the software understood while checking the hardware that one wheel was failed and that it needed to be taken out from the control loop. It takes 8minutes to the software to recognise a wheel permanent failure: this as in impact also on the choice of the number of maximum Timeouts and Retries that are allowed before executing NSM RACP set 2.

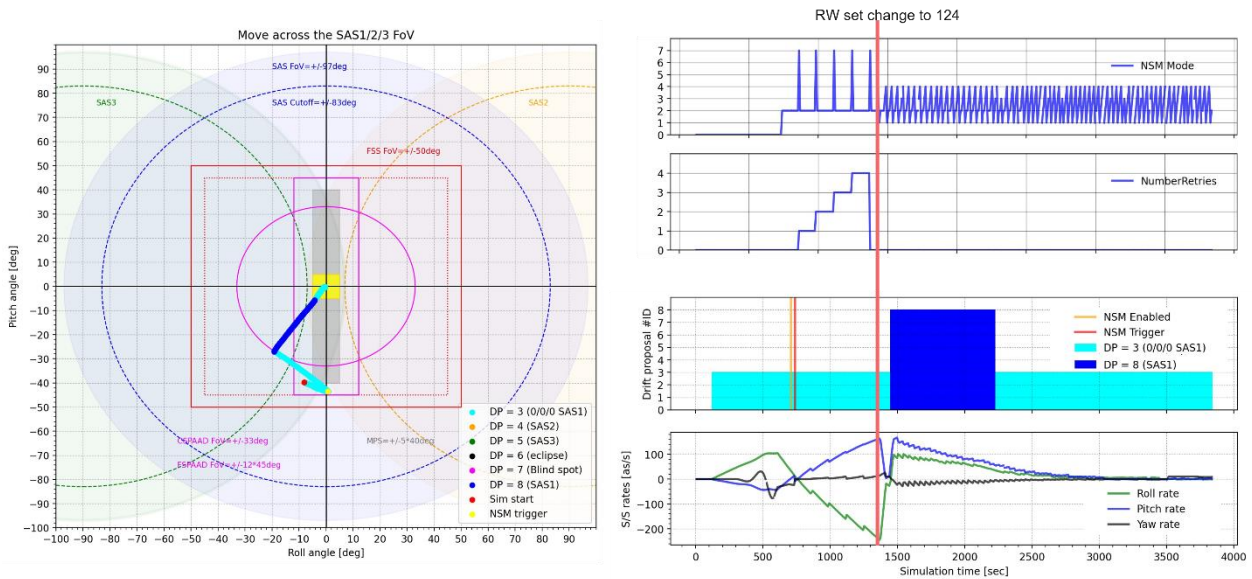


Fig. 22. RW3 permanent failure test case.

### 5. New Safe Mode, Current Status and Conclusions

After INTEGRAL thruster has been declared unreliable for RW angular momentum bias and after ESAM9, it was clear that thrusters could not be used anymore for Safe Attitude Emergency Mode. Consequently, ESAM was disabled, and the new safe mode was designed. A prototype, automated ground based NSM is operational since December 2021 and it is based on the Drift Proposal control algorithm. The on board NSM software was initially developed in a JavaScript prototype, to test it with the operational simulator and at a later stage it has been translated into ADA code, language compatible with the CDMU, Fig. 23.

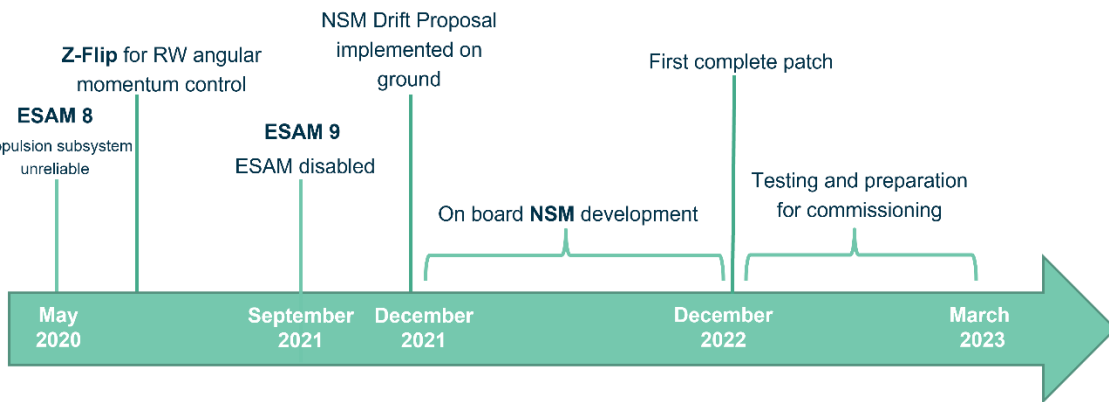


Fig. 23. Timeline from May 2020 until today.

The patch is also comprehensive of, Fig. 24:

- Code to extend usable memory to allow NSM software storage
- Code to extend Re-Action Control Procedure (RACP) from 50 TCs to 250 TCs, allowing AOCS and NSM configuration prior to NSM execution
- Code to read TM from ACC to command ACC (to handle interfaces)
- NSM software

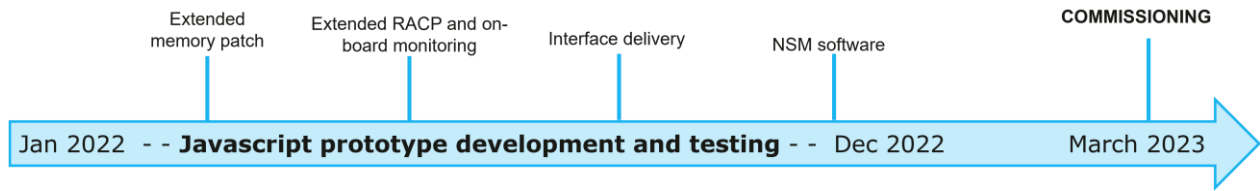


Fig. 24. Patch deliveries timeline.

NSM software configuration parameters are determined via extensive tests but not via optimisation: it is not excluded that in the future an optimisation study will be performed.

The NSM software formalises the concepts used to recover from ESAM9 and on the old Safe Attitude Emergency Mode. Its validation relies on wide range of failure scenario, where the conditions in term of unit availability, unit degradation, sun aspect angles, spacecraft and RWs angular momentum, ect. at the start of the recovery vary from being rather favourable to very extreme.

The NSM differs from old ESAM also with respect to the hardware used: indeed, IMUs readings are used to get rates information and thrusters are replaced with RW.

The INTEGRAL flight control team ran extensive tests on the NSM JavaScript prototype and on the software, patch delivered by industry. It is currently foreseen to upload and commission the NSM software patch in March 2023.

With NSM it is possible to reach Sun Pointing attitude within 40 minutes (average) and stable Sun pointing can be maintained for hours until the flight control team intervention. It is more precise and reliable with respect to the old ESAM (based on thrusters), more robust against permanent and temporary failures and will allow a significant reduction of mission costs due the lower need for ground station coverage. Its commissioning will be crucial for mission extension.

## References

- [1] C. Winkler, "The INTEGRAL mission," *A&A 411*, no. L1-L6, 2003.
- [2] G. R. J. P. Vedrenne, "SPI: The spectrometer aboard INTEGRAL," *A&A 411*, no. L63-L70, 2003.
- [3] P. L. F. Ubertini, "IBIS: The Imager on-board INTEGRAL," *A&A 411*, no. L131-L139, 2003.
- [4] N. Lund, "JEM-X: The X-ray monitor aboard INTEGRAL," *A&A 411*, no. L261-L268.
- [5] J. M. Mas-Hesse, "OMC: An Optical Monitoring Camera for INTEGRAL. Instrument description and performance," *A&A 411*, no. L261-L268, 2003.
- [6] E. Kuulkers, "INTEGRAL reloaded: spacecraft, instruments and ground system," *New Astronomy Reviews*, no. 93, p. 101629, 2021.
- [7] T. Godard, "The New Life of ESA's INTEGRAL Spacecraft Without Propulsion System," *Journal of the British Interplanetary Society*, vol. 75, no. 3, 2022.
- [8] "Three hours to save Integral," ESA, 18 10 2021. [Online]. Available: [https://www.esa.int/Enabling\\_Support/Operations/Three\\_hours\\_to\\_save\\_Integral](https://www.esa.int/Enabling_Support/Operations/Three_hours_to_save_Integral). [Accessed 26 08 2022].