Acta Astronautica 236 (2025) 417–431

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Towards the first European autonomous flight safety system- software and hardware design

Alejandro Sabán-Fosch [a,b,*], Eduard Diez-Lledó [a], Manel Soria [b], Miquel Sureda [b]

[a] GTD SSI, Passeig Garcia Fària, 17, Barcelona, 08005, Spain
[b] Department of Physics, Universitat Politècnica de Catalunya, 08222, Terrassa, Spain

ARTICLE INFO

Keywords:
AFSS
Flight safety
Spaceports
Launch vehicles
Reusability
System design

ABSTRACT

Flight Safety Systems (FSS) are responsible for the launcher neutralization, i.e. mission termination, based on the launcher and mission diagnostics. Existing systems require as input the launcher telemetry combined with independent tracking data (typically from radars) to properly and safely assess the launcher status. The assessment focuses on launch dynamics, while decisions regarding mission termination are delegated to the Flight Safety Officer (FSO) in the Mission Control Centre. Autonomous Flight Safety Systems (AFSS) are emerging developments capable of mitigating the spaceport infrastructure and the responsiveness limitations of current systems by evaluating termination decision without relying on humans and/or external ground tracking system data. This paper presents the software and hardware architecture, as well as the algorithms for real-time diagnostics and a deterministic decision-making process, for the first European AFSS prototype to be qualified in an operational environment. The qualification of the AFSS is conducted as part of the THEMIS stage hop flight tests at the Kiruna spaceport, scheduled for end-2025, within the Horizon Europe SALTO project. The design resulting from the Model-Based Systems Engineering (MBSE) approach shows how this prototype can be used in two different deployment scenarios: ground-based, replacing existing systems, or on-board in the launcher, eliminating the need for all spaceport infrastructures dedicated to FSS. The design also includes the characterisation of the system through a Reliability, Availability, Maintainability and Safety (RAMS) analysis and the prototype integration and testing. The result is a full functional AFSS for expendable and reusable launch vehicles, adaptable to different scenarios, safety rules, regulations and launchers. This work is intended to support European Flight Safety regulators and industry in the design and development of a multi-launcher AFSS for Europe.

Acronyms/Abbreviations

AFSS	Autonomous Flight Safety System
CF	Critical Function
CSG	Centre Spatial Guyanais
CNES	Centre National d'Études Spatiale
ECSS	European Cooperation for Space Standardisation
ELV	Expendable Launch Vehicle
ESA	European Space Agency
FE	Feared Event
FMEA	Failure Mode Effect Analysis
FMECA	Failure Mode Effect & Criticality Analysis
FTA	Failure Tree Analysis
FTS	Flight Termination System
FSS	Flight Safety System
FSO	Flight Safety Operator
GNSS	Global Navigation Satellite System
HIL	Hardware-in-the-loop
HW	Hardware
IIA	Instantaneous Impact Area
IIP	Instantaneous Impact Point
ILL	Impact Limit Line
IMU	Inertial Measurement Unit
IVHM	Integrated Vehicle Health Management
MBSE	Model-Based Systems Engineering
MTBF	Mean Time Between Failures
PIL	Processor-in-the-loop
RAMS	Reliability, Availability, Maintainability and Safety
REI	Regulations for the Exploitation of the Installations at CSG
RCC	US Range Commanders Council
RLV	Reusable Launch Vehicle

(continued on next column)

(continued)

(continued on next page)

* Corresponding author. GTD SSI, Passeig Garcia Fària, 17, Barcelona, 08005, Spain.
E-mail addresses: alejandro.saban@gtd.eu, alejandro.saban@upc.edu (A. Sabán-Fosch), eduard.diez@gtd.eu (E. Diez-Lledó), manel.soria@upc.edu (M. Soria), miquel.sureda@upc.edu (M. Sureda).

https://doi.org/10.1016/j.actaastro.2025.07.004
Received 29 November 2024; Received in revised form 26 May 2025; Accepted 3 July 2025
Available online 4 July 2025

(*continued*)

RT	Technical Regulations concerning authorisations for space operations
RTOS	Real-Time Operating System
SALTO	reuSable strAtegic space Launcher Technologies & Operations
SIL	Software-in-the-loop
SW	Software
TBC	To Be Confirmed
TBD	To Be Determined
TM	Telemetry
TRL	Technology Readiness Level
VLL	Vertical Limit Line
VTHL	Vertical Take-off and Horizontal Landing
VTVL	Vertical Take-off and Vertical Landing

1. Introduction

To ensure the safety of launch operations and mitigate risks to third parties in the event of a non-nominal and critical flight conditions, the implementation of a robust FTS is required for all launch vehicles. Conventional FSS rely on launcher telemetry data, ground-based radar tracking systems distributed around the globe, and an uplink to the on-board FTS for the termination command. The radar and telemetry data are processed by ground systems and displayed to the FSOs, who are responsible for issuing the termination command based on the violation of a set of flight safety rules defined by the spaceport flight safety regulatory authority. The termination command is then sent to the on-board FTS using the spaceport telecommand system. Such systems impose limitations on the range of trajectories available at a spaceport, due to the necessity for tracking and telecommand systems to be in line-of-sight of the launch vehicle. Furthermore, the current telecommand systems are limited to the transmission of a single signal, making it impossible to neutralize multiple objects, a necessity for RLVs. Lastly, the necessity for operators to undergo specific training and ground systems configuration for each mission requires a minimum time between successive campaigns.

In the context of New Space, two AFSS concepts have emerged as potential solutions to the limitations of existing FSS: ground-based AFSS (see Fig. 1a) and on-board AFSS (see Fig. 1b). Both systems operate autonomously, relying solely on data from the launcher, which eliminates the need for traditional ground infrastructure to monitor the flight, i.e. radar systems. The main distinction lies in the location of the equipment: at the spaceport (ground AFSS) or integrated directly into the launcher (on-board AFSS). The choice between these concepts depends on the specific safety regulations of the spaceport and the operational requirements of the launcher.

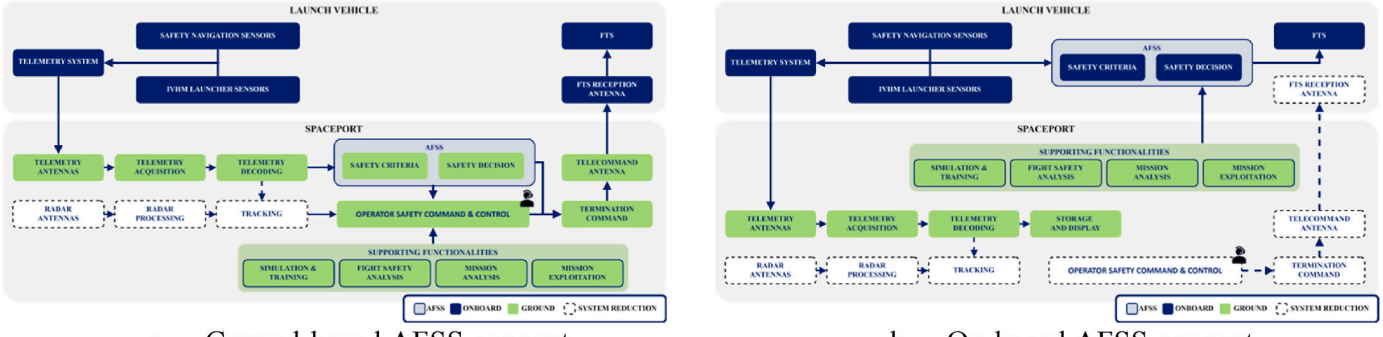
The ground AFSS strategy allows for the implementation of autonomous solutions at those spaceports where safety regulations do not allow the use of on-board AFSS. Additionally, this concept facilitates a gradual transition from conventional FSS to on-board AFSS by allowing both systems to operate in parallel, enabling thorough validation of the autonomous technology before full implementation with reduced hardware requirements.

Conversely, the on-board AFSS offers a significantly greater contribution to the advancement of New Space objectives by.

- **reducing and simplifying ground infrastructure.** The need for traditional radar systems is eliminated, telemetry processing systems are streamlined, and operator training systems become obsolete. This leads to a more efficient, compact, and cost-effective ground infrastructure;
- **minimizing ground operations.** Campaign preparation becomes more agile and responsive, requiring fewer operators at the launch base. This reduces overall mission preparation time and manpower needs, enhancing operational efficiency;
- **easing mission constraints.** The latency in safety-related decision-making is significantly reduced, as the current process -where signals travel from the launcher to ground operators and back to the launcher for neutralization-becomes obsolete. Consequently, ground safety limits and time-related constraints can be relaxed. The safety of the mission becomes independent of the telecommand line-of-sight and ground infrastructure, allowing for more flexible missions with multiple azimuths while enhancing overall safety;
- **facilitating the neutralization of multiple re-entry stages.** Current ground-commanded safety systems are unable to neutralize multiple objects in-flight due to the unreliability of simultaneous telecommands, which is prohibited in some locations. This limitation presents a challenge for RLV concepts, where several stages -such as a returning first stage and an ascending main stage require concurrent safety decisions. The on-board AFSS addresses this issue by implementing an independent safety system for each stage, allowing for safe and autonomous decision-making. Furthermore, the continuous health monitoring provided by AFSS enhances the safety of the re-entry process for reusable stages.

The result is both safer missions and a significant reduction in ground infrastructure and operational costs.

This paper summarizes the technical design work of the AFSS being developed by the authors under the Horizon Europe SALTO project contract. The current state of AFSS is presented in Section 2, followed by a discussion of regulations and requirements in Section 3. The MBSE software and hardware architectures are then described in Section 4 and evaluated through RAMS analysis in Section 5; the diagnostics and decision-making algorithms are described in Section 6. Finally, Section 7 covers the integration and testing of the ground AFSS version prototype to be used in the SALTO hop flight tests scheduled for end-2025 in Kiruna. With the real flight test, the project aims to reach a TRL 7 for both AFSS software and hardware, i.e. a model demonstration for operational environment [1]. This would result in the first AFSS qualified in real flight in Europe.



a. Ground-based AFSS concept. b. On-board AFSS concept.

Fig. 1. New Space AFSS solutions.

2. State of the art

At the time of this publication, there is no fully operational AFSS in Europe. Both public and private entities have conducted some developments in the last decade to have their own product without depending on export control regulations. The CNES, and the spaceports of Andøya Space (Norway) and Kiruna (Sweden) are the most ambitious players on that axe of innovation.

- CNES performed two iterations of their "*Kit Autonome comme Solution de SAuvegarde en Vol*" (KASSAV) for the Kourou spaceport, CSG. The first version of KASSAV, developed by Safran Data Systems and integrated by GTD, focused on the on-board determination of the launcher dynamics based on an IMU and the Galileo GNSS, thus providing launcher tracking along with the radar systems. This system has been qualified on-board the Ariane 5 VA253 [2]. During the second iteration of KASSAV, GTD performed a study to develop the diagnostics and decision-making software components to eliminate human intervention towards a fully autonomous safety system to be integrated in Safran Data Systems on-board equipment [3,4].
- the Scandinavian spaceports commissioned GTD to develop a semi-autonomous FSS, where the system can operate autonomously on the ground, but with the supervision of an FSO, and still relying on launcher telemetry data and radar tracking systems [5]. Both systems at the Kiruna and Andøya spaceports are already qualified and fully operational, awaiting the first signed micro launcher flights.

Other contributions in Europe to the AFSS include the announcement of the partnership between Sener Aeroespacial and Gilmour Space Technologies in 2022 [6] with the handicap of being a non-European system and hence not guaranteeing the independent access to space; the SAFEST Horizon Europe project, led by Sener Aerospacial, to develop an Autonomous Flight Termination Unit with TRL 5–6 [7,8] and hence not tested in an operational environment; and the results published by the Astos Solutions GmbH, HyImpulse Technologies GmbH and ArianeGroup consortium in mid-2022 and late 2023 under the ESA FTSNext contract for the design, development and testing of an AFSS but with a low TRL up to 4 [9–11].

On the other hand, in the United States of America (USA), there are two private companies using on-board AFSS: Space Exploration Technologies Corp. (SpaceX) and Rocket Lab USA, Inc. (Rocket Lab).

- In 2017, SpaceX achieved the milestone of being the first company to certify and deploy an on-board AFSS on a launch vehicle, effectively addressing the challenge of neutralizing multiple stages with telecommand systems for the Falcon Heavy RLV. This milestone was achieved using the Falcon 9 rocket on mission CRS-10 from Kennedy Space Center's Launch Complex 39A [12,13]. As stated in Ref. [12], SpaceX had previously employed the on-board AFSS in shadow mode alongside a conventional FSS to certify its system, i.e. enabling safety officials to issue a terminate condition.
- Rocket Lab conducted its first flight with AFSS from New Zealand in 2019 [14]. But it was not until 2023, that the company made its debut on U.S. soil. The launch was from NASA's Wallops Flight Facility in Virginia, with the Electron rocket using the AFSS developed by NASA, NASA Autonomous Flight Termination Unit (NAFTU) [15]. NAFTU is the result of the research, development and testing conducted by NASA in partnership with the US Department of Defense (US DoD) since 2000 [16,17], with the goal of providing industry with a configurable system adaptable to different launch vehicles and spaceport flight safety regulations, but restricted by US International Traffic in Arms Regulations (ITAR) export control regulations [18].

Finally, the Japanese space agency JAXA, in partnership with Space Engineering Development Co., Ltd, developed and successfully deployed an on-board AFSS in a sounding rocket in December 2023 [19,20].

3. Regulations and requirements

Regulations are fundamental to the design of any aerospace system; however, at the time of this publication, there are still no applicable regulations published for AFSS in Europe. It is acknowledged that CNES, as the major actor in Europe due to its responsibilities for the European Spaceport at CSG, is addressing this issue in the REI 2024 working document [21]. Therefore, pending the official publication of this regulation, the REI 2024 working document, the *LOI n°2018-518 concerning space operations* [22], the *Regulations for the Exploitation of the Installations at the CSG* (REI 2010-1) [23] and the *Technical Regulations concerning authorisations for space operations* (RT 2011) [24] should be considered as the baseline for developments concerning AFSS at CSG. On the other hand, Scandinavian spaceports specify in their safety manuals [25] that FSS shall satisfy the requirements defined in the US Range Commanders Council *Flight Termination Systems Commonality Standard* (RCC 319-19) [26], the *Global Positioning and Inertial Measurements Range Safety Tracking Systems Commonality Standard* (RCC 324-11) [27], the US Federal Aviation Administration *Part 147 Launch Safety* [28] and *Part 150 Launch and re-entry licence requirements* [29], and the *Flight Safety Analysis Handbook* [30]. This practice is also followed by other spaceports, such as those in United Kingdom, New Zealand and Australia [31–33].

The downside of directly considering the RCC 319-19 is that it imposes severe restrictions on the design and development of flight safety systems. In fact, the standard contains a comprehensive set of specifications for all components. As indicated in RCC 319-19, these limitations can be reduced by adapting the standard in accordance with the principles set forth in other less restrictive standards, such as the *European Cooperation for Space Standardisation* or the *DO-178C Software Considerations in Airborne Systems and Equipment Certification* [34]. **The decision to tailor the standard and adopt either the *ECSS* standards or the *DO-178C* is at the discretion of the spaceport safety regulator authority.** The *ECSS* standards that are of particular relevance in this context are the *ECSS-E-ST-10-06C Technical Requirements Specification* [35], the *ECSS-E-ST-10C Rev.1 System engineering general requirements* [36], the *ECSS-E-ST-40C Software Requirements* [37], the *ECSS-Q-ST-30C-Rev.1 - Dependability* [38], the *ECSS-Q-ST-30-02C Failure modes, effects (and criticality) analysis (FMEA-FMECA)* [39], the *ECSS-Q-ST-40C-Rev.1 Safety* [40], and the *ECSS-Q-ST-80C Software Product Assurance* [41].

An enhanced option to be compatible with any spaceport is to combine the previous tailored version of the RCC 319-19 with the REI 2010-1/2024 + RT 2011. However, it is important to be aware of the potential incompatibilities that may arise, as respecting one does not guarantee direct compatibility with the other (further discussed in Section 5.1). Consequently, when defining the requirements of the system, it is necessary to adhere to both regulations, or to the most restrictive of the two, in order to ensure compatibility with any spaceport, including CSG, which is currently the spaceport with the greatest potential to operate European micro-launchers and reusable concepts.

Accordingly, the system software and hardware requirements are defined by considering the above regulations, the state of the art (Section 2), and an analysis of customer needs, i.e. ArianeGroup SAS as SALTO project manager and Swedish Space Corporation as operator of the Kiruna spaceport. The requirements were also iteratively derived from the operational and system analysis as part of the MBSE process (Section 4).

4. System design

The following section presents a preliminary system design based on the identified requirements. The MBSE is implemented with the Architecture Analysis & Design Integrated Approach (ARCADIA) utilising the

Eclipse Capella software tool [42]. This method divides the engineering process into four phases to define needs (operational requirements analysis and system requirements analysis) and solutions (logical and physical architecture). The Eclipse Capella tool is used to develop a consistent system, software and hardware architecture. In conjunction with Capella, the use of the Enterprise Architect (EA) tool and the Systems Modelling Language (SysML) is preferred for a comprehensive understanding of the software definition, as it provides advanced features for detailing classes and software relationships [43]. To avoid uncorrelated design results from the models developed in both design tools, the Capella *SysML Bridge for Capella* add-on is used to synchronise and correlate both models [44].

4.1. System design drivers

The design of an AFSS for Europe must ensure compliance with regulatory and operational constraints while maintaining a modular and scalable architecture to support various launch vehicles, spaceport and mission profiles. A fundamental requirement, as established by safety regulations, is that the AFSS shall operate independently from the rest of the launcher's subsystems, including avionics, to ensure robustness against failures and interferences.

The key design drivers considered in this work are.

- **System Independence and Fault Tolerance:** The AFSS must operate autonomously, without reliance on the launcher's avionics or other subsystems, ensuring continued operation even in the event of vehicle failures. This includes independent processing, decision-making logic, and, where necessary, redundant communication links.
- **System Integration and Data Processing:** While independent, the AFSS must interface with launcher subsystems to receive essential flight data. It processes navigation and telemetry inputs through a unified interface while maintaining strict segregation from avionics control systems.
- **Safety and Reliability:** The AFSS must meet stringent reliability and safety standards, incorporating real-time fault detection, redundancy management, and fail-safe mechanisms. It is designed to autonomously execute safety actions based on predefined flight termination criteria.
- **Operational Flexibility:** The AFSS must support multiple operational modes, including pre-flight simulation, real-time monitoring, and post-flight analysis. It must be adaptable to various launch scenarios, including different flight corridors, spaceport locations, and mission requirements.
- **Computational and Algorithmic Performance:** The AFSS relies on advanced algorithms for trajectory prediction, violation detection, and termination logic. Its computational architecture must ensure real-time execution, deterministic response times, and the ability to manage multiple contingencies.
- **Regulatory Compliance and Certification:** The AFSS must conform to European and international safety regulations, including ESA and national space agency requirements. This necessitates rigorous validation and verification, including hardware-in-the-loop simulations and flight testing.

To focus on these critical aspects, the following assumptions and exclusions are made for the software and hardware design.

- Inputs from all launcher subsystems (including navigation sensors) are received from the same interface connection, and then distributed internally along the different components.
- The segregation and redundancy strategies and the equipment selection of the navigation sensors for position, velocity and attitude determination are not included within the scope of this AFSS design. These aspects will be addressed during AFSS integration with the launcher, considering factors such as mass and volume restrictions.
- The power supply system redundancy strategy is not included in the scope of this AFSS design.
- The termination system, i.e. pyrotechnics, engine shutdown managing, etc., is not included in the scope of this AFSS design, as it may differ depending on the launcher, e.g. Ariane 5 uses pyrotechnics, while Soyuz shuts down its engines.
- The specific physical interfaces with the launcher and spaceport, such as the avionics bus and the telecommand system, will be defined as TBD or TBC, as this will depend on the specific requirements of each client.

All excluded components are considered as external interfaces, provided that the software interfaces and data format presented in this work are respected. **The primary design effort is thus focused on tracking data processing, safety criteria, decision-making processes and the design of computational processors architecture.** Furthermore, the AFSS will be developed to operate independently, ensuring compliance with safety regulations that mandate segregation from launcher avionics and other subsystems.

4.2. System analysis

The system analysis captures the functionalities of the system under consideration. In this context, the functionalities common to both ground-based and on-board AFSS concepts are identified below. These shared functionalities reflect the core contributions of the system, independent of implementation specifics, and provide a foundation for the subsequent architectural and design activities.

a. An **acquisition** and **pre-processing function** that retrieve data from dedicated safety sensors. To accommodate different AFSS concepts, the acquisition should be transparent to the source of the data, which may come directly from the sensors, the launcher bus or the telemetry antennas at the spaceport. This transparency can be ensured as long as the software interface and the data format are respected. Relating to the pre-processing function, it includes the generation of two independent data streams, which enables safety operators to disable any of the diagnostics modules if necessary,
   i. a flight dynamics diagnosis data stream including navigation data;
   ii. and a Vehicle Health and Monitoring data stream including all the sensor required to determine the health of the launcher.

b. A **flight dynamics diagnosis function** to assess the status of the launcher dynamics based on the current flight phase and a set of flight safety criteria. Such criteria include,
   i. Tracking Sources health assessment, based on their operational status;
   ii. Hybrid Navigation Solution health assessment, focusing on the numerical stability of the covariance matrix, i.e. checking semi-positiveness and symmetry matrix conditions. The Hybrid Navigation module integrates data from various sensors, e.g., IMU and GNSS, to estimate position, velocity and attitude, which are then used for subsequent diagnostic functions (iii), (iv), and (v). As the selection of navigation sensors is outside the scope and treated as an input to the AFSS, their key parameters -such as acquisition frequency, noise characteristics, bias, and scale factors-are parametrized to accommodate a variety of sensor configurations;
   iii. Flight path integrity check, where the current position is compared against predefined horizontal and vertical safe corridors to ensure the vehicle remains within acceptable limits;
   iv. Impact Predictor check, where the impact point and its uncertainty area are computed and compared to predefined safe impact zones to assess potential risks.

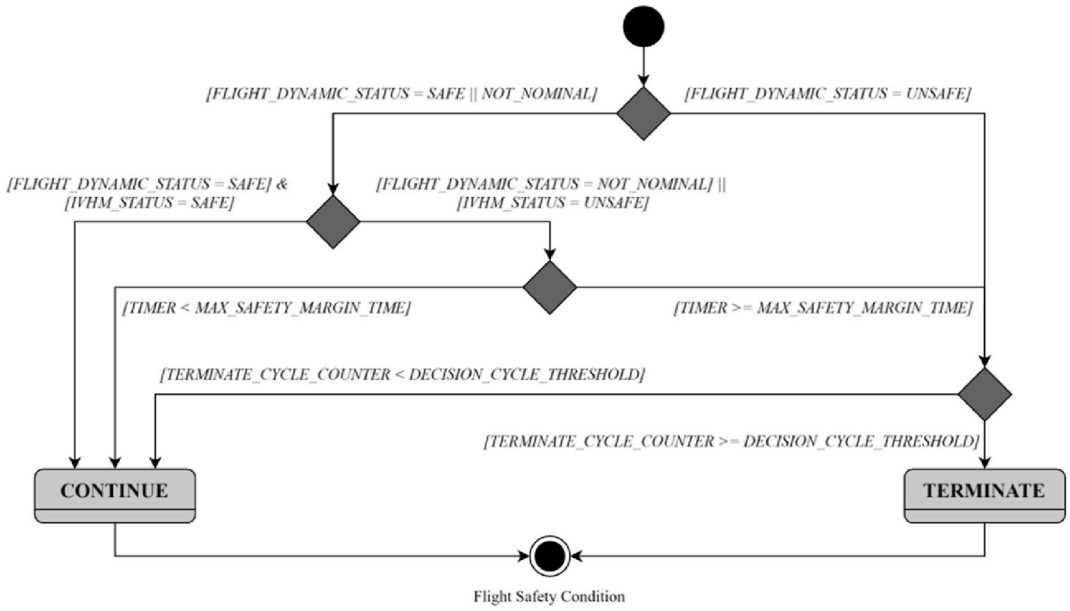


Fig. 2. AFSS Flight Safety Condition assessment logic.

For each flight phase, a degree of influence for each safety criteria in the evaluation of the status of the flight dynamics is configured. The result of the assessment is one of the following status: SAFE, NOT_NOMINAL or UNSAFE. Furthermore, this function also returns a safety margin timer, referred to as "green time", which indicates the amount of time required for the launcher to violate any of the aforementioned criteria.

In order to facilitate the configuration of the AFSS, allowing for a rapid adaptation to multiple launcher concepts, and multiple spaceports and missions, the transition between phases is addressed through the use of an interpreter capable of reading literal logical expressions indicating the thresholds of launcher dynamics variables, such as ground distance from the launchpad, launcher acceleration or staging phase. Furthermore, all the predefined safety parameters, such as safe impact regions and safe corridors, are characterised by a parametrisation that facilitates configuration for each mission.

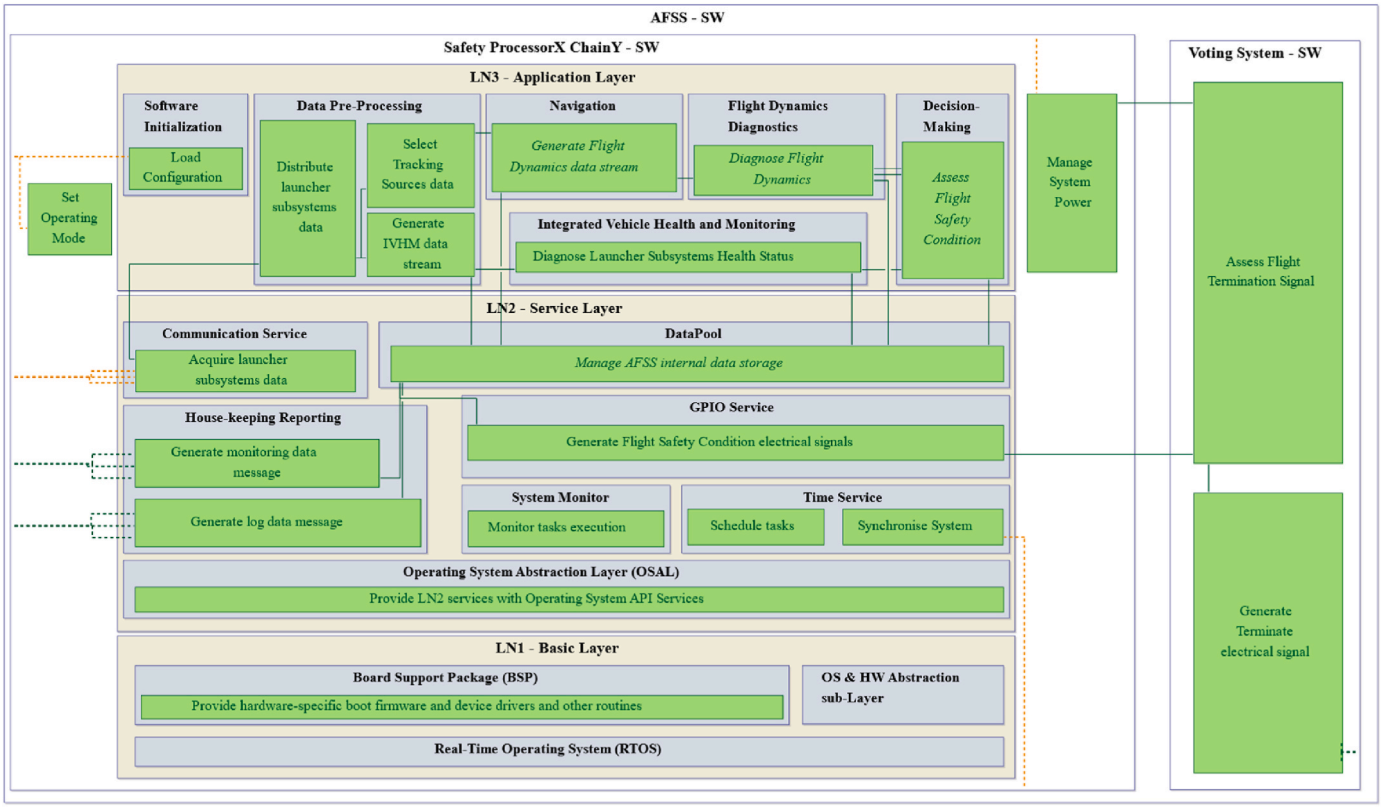


Fig. 3. AFSS logical components and functions allocation.

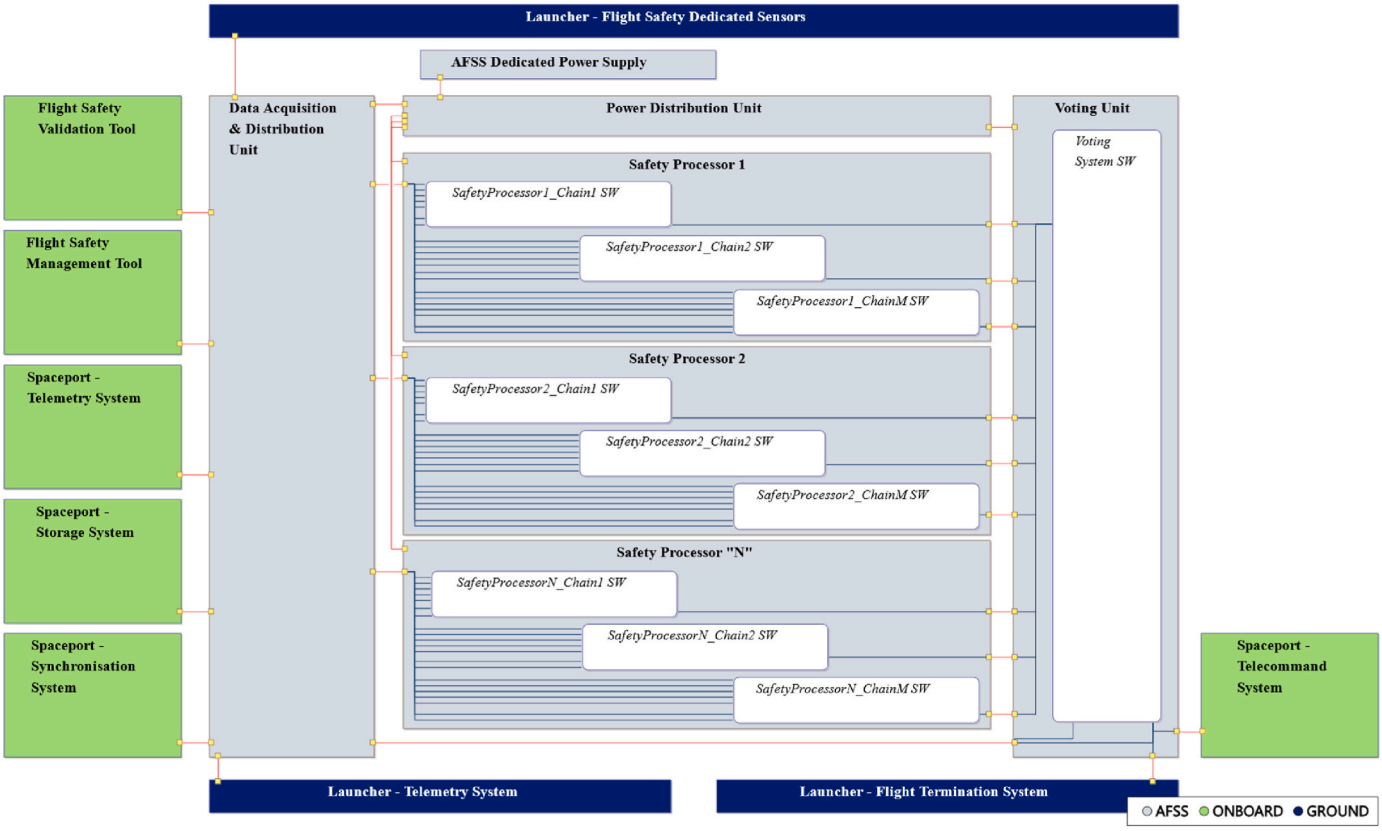


Fig. 4. AFSS physical components and logical components allocation.

c. An **integrated vehicle health status and monitoring (IVHM) diagnosis function** to assess the overall health status of the vehicle based on the health condition of its critical subsystems, i.e. propulsion, avionics, actuators, etc., and their impact on the overall safety of the launcher. The assessment is done using a configurable criticality table that indicates the impact of each launcher subsystem on mission safety. Its configurability is achieved through the utilisation of the identical flight phase segmentation methodology employed in the flight dynamics diagnosis function. The resulting condition is a single launcher health status: SAFE, NOT_NOMINAL or UNSAFE.

In order to meet the spaceport regulations and the launch operator requirements, the use of this function can be enabled or disabled at the configuration phase. The configuration also includes the selection of which subsystems should be monitored.

d. A **flight safety condition assessment function** that returns a CONTINUE or TERMINATE status based on the flight dynamics and the vehicle health diagnosis, and a "green time" (see Fig. 2). The "green time" is a safety margin timer which enables the AFSS to await a specified interval for the launcher to recuperate from a degraded but non-hazardous state. The function allows operators to limit the maximum timeframe to a pre-defined value.

e. A **flight termination signal generation function** to trigger the termination signal to the FTS based on the status of the flight safety condition assessment. This functionality is isolated from the flight safety condition assessment function to facilitate the implementation of redundant software and hardware solutions.

Further functionalities such as **operator authentication**, **operating mode selection**, **system power management**, **configuration loading**, **system synchronisation** and **tracking source data fusion (hybrid navigation module)** are also identified during this design phase.

4.3. Logical architecture

The logical architecture outlines how the system functionalities identified in the System Analysis (see Section 4.2) are implemented. The allocation of logical components and functions shown in Fig. 3 identifies two principal components, each further decomposed into multiple sub-logical components. These sub-components encapsulate the system functionalities outlined in Section 4.2.

- Safety Processor SW: responsible for assessing the flight safety condition. It implements functions a) to d) outlined in Section 4.2.
- Voting System SW: responsible for triggering the terminate signal. It implements function e) outlined in Section 4.2.

This structured decomposition not only clarifies the architecture of the system but also ensures scalability, modularity, and consistency, providing a robust framework for detailed technical and physical designs.

Finally, to further enhance modularity, reusability, and maintainability, the implementation of these functions adopts a standard software three-layer strategy, as described below.

- The **Application Layer (LN3)** includes the processing and computation functions described in Section 4.2, encompassing all application-specific logic.
- The **Services Layer (LN2)** provides general-purpose data handling services, including the Data Pool, Communication Service, Time Service, and General-Purpose Input-Output Control Service. It facilitates communication between the LN3 and the LN1, ensuring smooth integration.

- The **Basic Layer (LN1)** contains the RTOS and the Hardware Abstraction sub-Layer, providing low-level hardware control and resource management.

4.4. Physical architecture

Finally, the physical architecture allocates the logical components identified in Section 4.3 and their associated functionalities to hardware equipment. The Physical Architecture, presented in Fig. 4, builds on the Logical Architecture by allocating logical components and their defined interactions to hardware, software, or a combination of both. While it does not focus on specific equipment or implementation technologies, it defines the system in terms of generic hardware and software specifications, thereby ensuring flexibility and adaptability in the subsequent design and development stages (see Section 7 for SALTO use case specific hardware and software specifications).

As evidenced in Fig. 4, the physical architecture considers the implementation of multiple instances of the Safety Processor SW logical component alongside multiple Safety Processor hardware components. Consequently, the Voting Unit should generate the flight termination signal by resolving the multiple flight safety conditions issued by each Safety Processor SW instance through a consensus-based approach. This design approach leads to.

- an increase in the reliability and availability figures of the system;
- a reduction in development costs, as redundant and independent software implementations can relax coding standards requirements for the Safety Processors. This topic has been fully addressed by CNES in RNC-CNES-Q-HB-80-507 (p. 17 of 41, Section 7.5.2) [45].

This redundancy-based design enables the code criticality to be reduced. Nevertheless, it is only valid if different implementations are produced in the physical components and different data sources are used. It is thus necessary that the pre-processing and diagnosis functions employed in the different software instances must be different, e.g. different Kalman filters and different impact predictor computation algorithms. Furthermore, ensuring different input data sources, such as distinct IMU and IMU + GNSS combinations, is critical for achieving effective redundancy. It is also important to acknowledge that, according to ECSS-Q-ST-30C Rev.1 (Tables 5–3, p.23) [38], this approach is not deemed to be valid for ESA. **Consequently, the above architecture should be subjected to a review in accordance with the applicable regulations in the specific use case, i.e. determining whether CNES or ESA regulations are applicable.**

The selection of the number of processors (“N") and processes (“M") should be guided by the adopted system design strategy and informed by the results of a RAMS analysis (see Section 5) tailored to each use case. In regard to the design strategy, two primary approaches have been identified for consideration: (i) to minimise equipment and software, resulting in a reduction in mass and software costs, but increasing the MTBF requirements of the equipment and coding rules constraints; or (ii) to reduce the MTBF requirements of the equipment and software coding requirements, thus reducing the hardware and software costs, but increasing overall mass.

Finally, the segregation of non-safety-critical functions is also considered in the design, as this represents a cost-effective strategy for reducing development and qualification costs. This segregation involves extracting these functions from the primary process to a secondary one with fewer coding restrictions, particularly when these functions do not directly impact mission safety. In the context of the AFSS, configuration loading and the telemetry processing functions can be segregated since they do not influence the safety of the mission. In the case of the former, if the safety configuration is not loaded correctly, the flight safety team will abort missions before tH0 (launch time); in the case of the latter, the worst-case scenario is that no real-time data can be recovered for post-mission analysis. Separating configuration loading from the source code also eliminates the need to re-qualify the entire software before each flight. Nevertheless, as set forth in Article 63.2 of the CNES working document REI 2024 [21], CNES does not accept the loss of telemetry data, as it must be available to the launch campaign director to enable him or her to make an independent decision regarding the termination or continuation of the flight. **Therefore, in order to implement this AFSS at the European Spaceport at CSG, it would be advisable to consider a version of the proposed system that does not segregate the telemetry function.**

5. Reliability, Availability, Maintainability and Safety (RAMS) system analysis

The following section presents a qualitative RAMS analysis of the system presented in Section 4. This qualitative analysis consists of: (i) a regulatory requirements analysis; (ii) a Failure Mode Effect and Criticality Analysis; and (iii) a Failure Tree Analysis.

5.1. RAMS regulations requirements analysis

As stated in Section 3, an AFSS system expected to be used in multiple launchers and multiple spaceports must comply with the specifications set forth in the REI 2010-1/2024+RT 2011 and the RCC 319-19 standards. In this regard, the standards specify the following RAMS metrics,

> “*Article 20: … a) Launch risk: 2e-5 for the entire launch phase, taking into account launch system degradations and including the fallout of components expected to detach from the launcher without being placed in orbit. …*”. *(RT 2011, p.7 of 17)* [24].

> *“3.2.2. Reliability. An FTS shall have a statistically predicted reliability with a 95 % single-sided lower confidence boundary of at least 0.999 …”. (RCC 319-19, p. 36 of 549, Section 3.2)* [26].

As evidenced by the definitions, RT 2011 defines reliability in terms of launch mission, whereas RCC 319-19 directly defines the reliability of the FTS system. Therefore, to compare the two standards, the definition of reliability in RT 2011 should be broken down into the key systems involved in a launch mission until AFSS reliability is derived. Note that reliability is the complement of the probability of failure.

The REI 2010-1/2024, considers two phases in a launch mission: *(i)* “*Mission de Sauvegarde et d’Intervention, MSI” (Article 63, p. 53 of 74)* [21]; and *(ii)* “*Mission de Sauvegarde et Alerte, MSA*” *(Article 64, p. 55 of 74)* [21]. The MSI focuses on assessing and mitigating risks associated with a launch vehicle, including potential interventions to neutralize it if necessary. This mission begins when the vehicle leaves the ground and ends at the latest the moment its impact area touches the territorial sea of the first State encountered outside French Guiana. In contrast, the MSA is dedicated to real-time monitoring from lift-off, ensuring the vehicle operates correctly and promptly alerting relevant authorities in the event of a failure. This oversight extends through the deorbiting of the final stage and, in cases of controlled re-entry, continues until the designated landing zone is assessed. Therefore, the AFSS, is only involved in the MSI launch mission phase and thus this phase will be the focal point of the present discussion.

During the launch mission MSI phase, the probability of causing death(s) can be caused by two mutually exclusive scenarios: (i) the launch vehicle fails and the AFSS system does not trigger the terminate condition; or (ii) the launch vehicle fails and the neutralization system does not perform the termination action. It should be noted that the events described are independent, as the outcome of one does not affect the outcome of the other events.

$$P(k \geq 1, MSI) = P(lau.\,fails)[P(AFSS\,fails) + P(AFSS\,succ.)P(neutr.\,fails)] \quad (1)$$

Where.

Table 1
Feared Events analysis.

ID	Feared Event	Severity of consequence			Ref. to Critical Function - CF-N -
		tH0 - N	tH0 + N *shadow*	tH0 + N *autonom.*	
FE-1	AFSS is not available				
FE-1.1	All Safety Processors fail	SOP-1	SOP-5		CF-0
FE-1.2	The Voting Unit fails	SOP-1	SOP-5		CF-0
FE-1.3	The Power Distribution Unit fails	SOP-1	SOP-5		CF-0
FE-1.4	The Data Acquisition and Distribution Unit fails	SOP-1	SOP-5		CF-0
FE-2	AFSS incoming data error				
FE-2.1	AFSS incoming data source error				CF-1
FE-2.2	AFSS incoming data is not available at AFSS		SOP-4	SOP-4	CF-1
FE-2.3	AFSS incoming data processing error				CF-1
FE-2.4	AFSS incoming data processing freezes		SOP-4	SOP-4	CF-1
FE-3	Termination failure				
FE-3.1	Navigation hybridisation failure				CF-2
FE-3.2	Impact check criterion failure		SOP-5,6		CF-3
FE-3.3	Flight path check criterion failure		SOP-5,6		CF-3
FE-3.4	IVHM criterion failure		SOP-5,6		CF-4
FE-3.5	Decision-making failure		SOP-5		CF-5
FE-3.6	Voting system failure		SOP-5		CF-6
FE-4	Unnecessary termination				
FE-4.1	Navigation health status check false positive	SOP-3	SOP-5		CF-2
FE-4.2	Impact check criterion false positive	SOP-3	SOP-5		CF-3
FE-4.3	Flight path check criterion false positive	SOP-3	SOP-5		CF-3
FE-4.4	IVHM criterion false positive	SOP-3	SOP-5		CF-4
FE-4.5	Decision-making false positive	SOP-3	SOP-5		CF-5
FE-4.6	Voting system false positive	SOP-3	SOP-5		CF-6
FE-4.7	Hardware sends spurious termination command	SOP-3	SOP-2		CF-6
FE-5	Failure to send AFSS data to FSO.				
FE-5.1	AFSS outgoing data processing freezes	SOP-1,3	SOP-5		CF-7
FE-5.2	Failure to communicate with FSO display	SOP-1,3	SOP-5		CF-7
FE-6	Failure to command terminate				
FE-6.1	No termination command is sent to the FTS				CF-6
FE-6.2	Spurious termination command sent to the FTS				CF-6

- $P(k \geq 1, MSI)$: probability of causing "k" deaths during MSI launch mission phase.
- $P(lau.\ fails)$: probability of the launcher to fail.
- $P(neutr.\ fails)$: probability of the neutralization system to fail in executing terminate action.
- $P(AFSS\ fails)$: probability of the AFSS system to fail in triggering terminate condition.
- $P(AFSS\ succ.)$: probability of the AFSS to succeed in triggering terminate condition. It is the complement of $P(AFSS\ fails)$.

From Eq. (1), it can be observed that the reliability figures in the RT 2011 standard are highly dependent on the reliability of the launcher, whereas this is not the case in the RCC 319-19. Thus, be compliant with RCC 319-19 will not guarantee the compliance with REI 2010-1/2024+RT 2011 regulations. This conclusion has a significant impact on the design of a system that must adapt to any launcher and be REI 2010-1/2024+RT 2011 compliant, since the higher the launcher reliability, the fewer requests will be made to the AFSS, and vice versa. For this reason, a specific analysis must be performed for each launcher and spaceport, with implications for the selection of AFSS hardware equipment and redundancy policies.

Furthermore, Eq. (1) highlights that the neutralization system has its own probability of failure due to the assumption made in Section 4.1, where the AFSS operates independently of the termination system. This approach ensures that the AFSS remains adaptable to various launchers and termination mechanisms. For instance, Ariane 5 relies on a pyrotechnic destruction system, whereas Soyuz shuts down its engines and allows the vehicle to fall.

Finally, it is worth noting that a more conservative approach could be adopted, wherein the 2e-5 probability of failure from the RT 2011 standard is directly assigned to the AFSS, bypassing the detailed probability analysis presented earlier. However, this simplification might lead to unnecessary higher AFSS costs for launchers with higher reliability, since the AFSS will has been designed with overestimated reliability figures.

5.2. Failure Mode Effect Analysis (FMEA)

The initial step in developing an FMECA analysis is to identify the potential failure modes (FM) and their associated effects, which are also

Table 2
Standard Operational Procedures hypothesis.

ID	Standard Operational Procedure (SOP)
SOP-1	The FSO shall halt the launch countdown if the AFSS fails before the launch.
SOP-2	Any AFSS termination command shall be reinforced with a TERMINATE via the FTS control panel.
SOP-3	Safety barriers shall be put in place until tH0. A spurious terminate command at (tH0-N) will not produce launcher destruction.
SOP-4	If there is a loss of (N) seconds of AFSS incoming data, the AFSS shall terminate.
SOP-5	If an UNSAFE FLIGHT alarm is generated from any cause, the FSO shall terminate. Potential causes of unsafe flight alarm are: - AFSS incoming data outage > (N) sec.; - Flight Dynamics data exceeds Impact or flight path safety limits; - IVHM UNSAFE status condition > "green time" seconds; - AFSS critical system failure.
SOP-6	The Impact check criterion shall apply throughout the flight from tH0 until the launcher has achieved the safe flight condition defined in the flight safety plan.

referred to as feared events. This is achieved through an FMEA analysis.

Table 1 presents the feared events (FE) identified for the AFSS system presented in Section 4 (to both shadow and autonomous modes), resulting from the FM analysis. Each FE is assigned a consequence severity level in accordance with the definitions established in Section 5.3: *Safety Critical* (red), *Mission Critical* (yellow), *Non-critical* (green) and *Not Applicable* (grey). It should be noted that the FE presented may be derived from lower-level failure modes, which for reasons of simplicity are not shown in this article.

It is evident that the level of severity associated with each event can be mitigated through the implementation of operational procedures designed to address potential safety threats to the mission. The operational procedures that were applied in the case study presented in this article are outlined in Table 2.

5.3. Criticality analysis

In order to complete the FMECA analysis and progress from FMEA to FMECA, it is necessary to conduct a criticality analysis. The criticality analysis allows to identify the risk associated to each component comprising a system. This identification permits to stablish development constraints, such as redundancy policies, coding categorisation and specification relaxation assumptions, such as those discussed in Section 4. According to the regulations introduced in Section 3, the consequences of a failure, i.e. the risk, may be classified into four categories.

- Safety Critical: A failure of this function could result in serious injury or fatality and there is no failure mitigation or alternative system functions.
- Mission Critical: A failure of this function may cause mission failure, but,
  a. cannot be a direct cause of serious injury or fatality;
  b. May only cause serious injury/fatality if an alternative system function also fails.
- Non-critical: A failure of this function may cause mission delay but may not cause loss of the mission.
- Not Applicable: The function is not applicable in this phase of the mission.

The AFSS functions identified in Section 4.2 are presented in Table 3 along with the severity levels for the availability and integrity quality

Table 3
Critical Functions analysis.

ID	Critical Function (CF)		Availability (severity)	Integrity (severity)
CF-0	AFSS operational availability.	*ground*	*Mission critical*	*Not Applicable*
		on-board	*Safety critical*	
CF-1	To acquire AFSS incoming data.		*Mission critical*	*Safety critical*
CF-2	To compute navigation solution, and check tracking sources and navigation solution health status.		*Safety critical*	*Safety critical*
CF-3	To diagnose Flight Dynamics.		*Safety critical*	*Safety critical*
CF-4	To diagnose Vehicle Health status.		*Mission critical*	*Mission critical*
CF-5	To assess Flight Safety Condition.		*Safety critical*	*Safety critical*
CF-6	To command TERMINATE signal.		*Safety critical*	*Safety critical*
CF-7	To communicate with FSO Display.		*Mission critical*	*Mission critical*

Table 4
Logical components criticality assignment.

Critical Function / SW Components	CF-0	CF-1	CF-2	CF-3	CF-4	CF-5	CF-6	CF-7
LN1	X	X					X	
LN2	X	X	X	X	X	X	X	X
LN3 / SW Initialization	X							
LN3 / Acquisition and Pre-processing function		X						
LN3 / Navigation			X					
LN3 / Flight Dynamics diagnosis				X				
LN3 / IVHM diagnosis					X			
LN3 / Decision-making						X		
LN3 / Voting							X	

Table 5
Physical components criticality assignment.

HW Components \ Critical Function	CF-0	CF-1	CF-2	CF-3	CF-4	CF-5	CF-6	CF-7
Flight Safety Processor(s)	X	X	X	X	X	X		X
Voting Unit	X						X	X
Data Acquisition and Distribution Unit	X							
Power Distribution Unit	X							

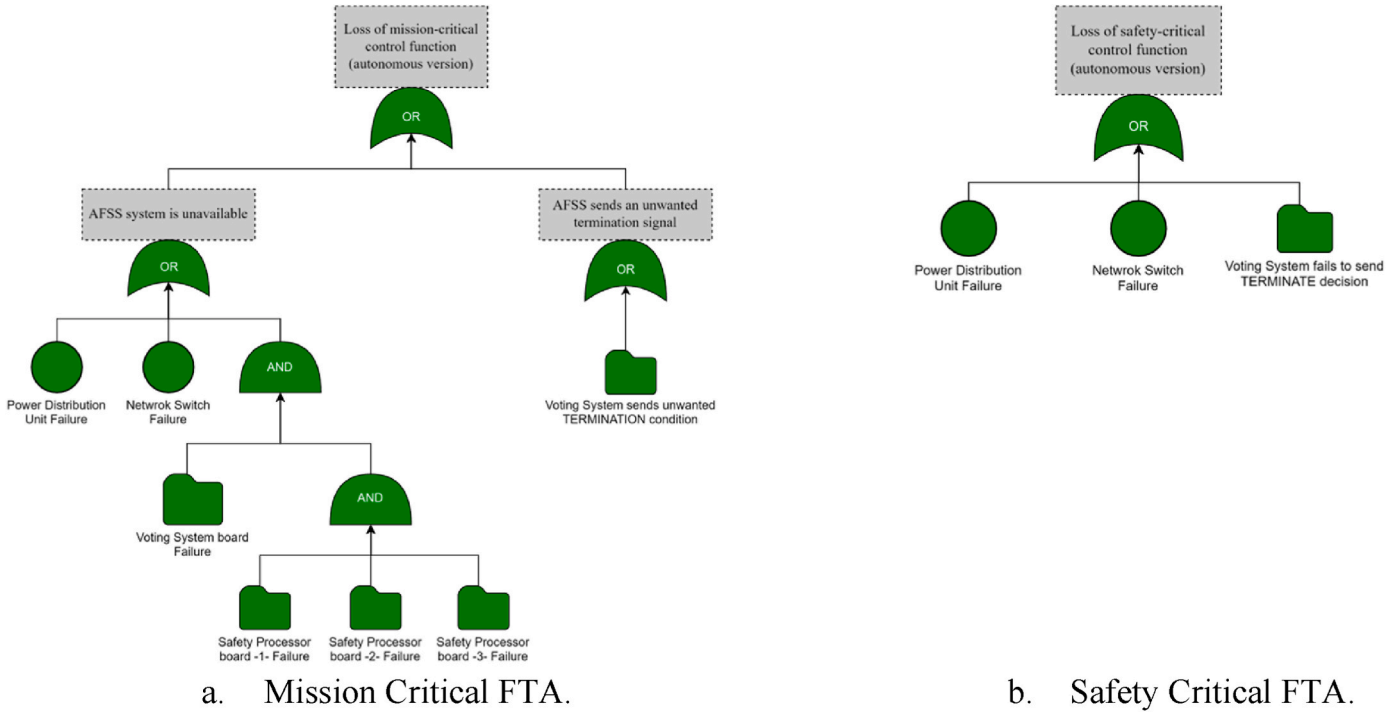


Fig. 5. Failure tree analysis.

metrics. In order to assign a severity level to each critical function (CF), it is first necessary to identify which CF is affected by which FEs identified in the FMEA (see Table 1). Once the assignment has been made, the severity level of the CF is then assigned in accordance with the severity level of the worst FE identified. Note that the severity level assigned to a CF may vary depending on the specific AFSS concept that is in place, such is the case of the CF-0.

As stated in Section 4.3 and Section 4.4, AFSS system functions are allocated to logical and physical components. Therefore, Tables 4 and 5 allocate the severity levels to both logical and physical components. It is important to note that, in Table 4, the logic components LN1 and LN2 group their respective subcomponents, as defined in Section 4.3; in contrast, LN3 logic component is broken down. This simplifies the table as all subcomponents of LN1 and LN2 contribute to the same CFs with the same level of criticality. As expected, most of the AFSS components are Safety Critical. This result implies higher demands for both the software coding rules and the selected hardware equipment. However, as already discussed in Section 4.3 and Section 4.4, redundant and independent implementations can reduce these solicitations. Furthermore, the segregation of non-safety critical functions is also discussed in Section 4.4 with the objective of reducing development costs. This segregation entails the extraction of these functions from the primary process to a secondary one with fewer coding restrictions.

5.4. Failure Tree Analysis (FTA)

Finally, to complete the RAMS analysis, the FMECA outcomes should guide the FTA to estimate the probability of failure for each equipment. The fault trees represent the failure modes and feared events identified in the FMEA, while Criticality Analysis can be used to categorized these events. Then, the FTA assigns a quantitative probability of failure to more accurately assess system-level risks, enabling the assignment of quantitative quality requirements to the equipment. In this process, only hardware is considered, as the software is not considered to fail. This is on the basis that the software is designed in accordance with the relevant coding standards and that the relevant validation procedures have been carried out.

Fig. 5 presents the FTA of both *Mission-Critical* and *Safety-Critical* cases. The events of the sub-diagrams -folder icons-in Fig. 5 are defined as follows.

- Safety Processor boards and Voting Unit failures can be caused by: (i) a processor chip failure; or (ii) a RAM memory failure; or (iii) a network interface failure; or (iv) a power rails failure.
- Voting Unit TERMINATION signal failures (unwanted condition or inability to send the signal) are associated with: (i) the inability to read Flight Safety Conditions electrical signals caused by the General-Purpose Input-Output pins interface with the Safety

Processor(s), or by the incapacity of the Safety Processor(s) to generate the electrical signals; or (ii) a failure of the Voting Unit.

6. Algorithms

The algorithms included in the LN3 SW Application Layer are the fusion of navigation sensors, the diagnostic algorithms for calculating the Instantaneous Impact Point (IIP), Impact Area (IIA) and Impact Status assessment, and the decision-making.

- Hybrid navigation algorithms (fusion of navigation sensors): different Kalman filters have been studied and implemented by the authors in previous work to fuse data coming from IMU and GNSS navigation sensors [46,47], thus improving the availability and the robustness of the system against external threats such as GNSS jamming. These studies include the use of the Extended Kalman Filter (EKF) and the Indirect Kalman Filter (IKF) algorithms. In Ref. [46], the EKF and the IKF were proposed to work sequentially to improve the navigation solution in transient phases (e.g. stage separation, engine re-ignition, etc.), as the IKF performs better in highly non-linear scenarios. In addition, the Unscented Kalman Filter (UKF) is also considered as an improvement in the navigation algorithm library, as it combines the advantages of the EKF and the IKF, but at the expense of more demanding computational requirements [48].
- IIP and IIA algorithms: F&G series algorithm has been studied, implemented and successfully tested under real-time conditions by the authors in Ref. [49]. This algorithm is based on a Taylor decomposition of the atmospheric ballistic fall model equations, so it can reduce the number of iterations required to compute the instantaneous impact point. In addition, the authors in Ref. [49] also propose two methods to compute an uncertainty region around the IIP, i.e. the IIA, based on the covariance matrix of the navigation solution: (i) a linear propagation method; and (ii) a non-linear propagation method based on the Scaled Unscented Transform (STU) mathematical transformation. Besides the F&G series, the IIP algorithm library also includes a non-iterative IIP calculation method based on Keplerian equations [50], and a plane-Earth parabolic trajectory model with first-order corrections for surface curvature, gravity variation and Earth rotation [51], to ensure differences in algorithm implementation between the safety chains.
- Impact Status assessment: to evaluate whether the IIP is within safe regions, the winding number algorithm described in Ref. [52] to solve *Point-In-Polygon* problem is used. On the other hand, for the IIA status assessment, the percentage of intersection area is used instead. The computation of the intersection area between the IIA and a safe region is accomplished by using the Commercial Off-the-Shelf (COTS) *General Polygon Clipper* (GPC) software library [53]. The use of the IIP or the IIA assessment depends on the directives given by the spaceport Safety Regulator Authority. It is therefore incorporated into the AFSS configuration parameters.
- Decision-making algorithm: Fuzzy Logic decision-making theory has been studied and implemented by the authors in Ref. [54], but the feedback received from the stakeholders indicates that a deterministic decision algorithm should be used rather than using Fuzzy Logic or Artificial Intelligence approaches. Another point raised is that the algorithm should be adaptable to the different phases of the mission. In order to achieve these objectives, the decision-making algorithm has been designed in two stages by the implementation of the functions discussed in Section 4.2: (i) the flight dynamics health status assessment; and (ii) the flight safety condition assessment.

7. Software/hardware integration and testing

In order to achieve the objective of reaching a TRL 7 for both software and hardware, it is necessary to test the AFSS in a real operational environment. Accordingly, the authors are developing a prototype of the AFSS based on the design outlined in Section 4. This prototype is scheduled to be deployed during the THEMIS flight tests in Kiruna. Since the prototype is intended for ground use, it does not need to meet the vibration and space environment requirements, significantly reducing equipment costs. However, ensuring the reliability of the prototype remains critical. For this reason, a redundancy strategy using three safety processors, each executing three instances of the Safety Processor SW, has been implemented, i.e. a total of nine safety chains. This decision is based on the findings of the RAMS analysis conducted by the authors for this use case, which considers the hardware equipment listed in Table 6 and the use of RTEMS [55] as the Real-Time Operating System (RTOS). Additionally, as suggested in the RTEMS documentation, the open-source Das U-Boot bootloader [56] is used to perform low-level hardware initialization tasks and to boot the BeagleBone Black RTEMS kernel.

Table 6
List of hardware equipment utilized in the AFSS prototype.

HW Equipment	Units	Relevant Features	Physical Component from Section 4.4
BeagleBone Black board [57]	4	- 1 GHz single core ARM Cortex-A8. - SDRAM: 512 MB DDR3L 800 MHz. - 4 GB, 8 bit Embedded MMC (eMMC). - GPIO Interfaces via 2 × 46 pin header. – 10/100M Ethernet interface. - Power Source: 5VDC@2A	x3 - Safety Processor(s) x1 - Voting Unit
D-Link Switch	1	– 8 Gigabit Ethernet ports.	Data Acquisition and Distribution Unit
Phoenix Power Distributor	1	- Power distributor PTRV Red (VCC). - Power distributor PTRV White (GND). - 2A Fuse (x5)	Power Distribution Unit
RS Pro Power adapter	1	– 230VAC-5VDC	Power Distribution Unit

Concerning the software development, all the functions described in Section 4 are implemented using the C/C++ programming language, in accordance with the MISRA C++:2008 coding standard [58], and the results of the software criticality RAMS analysis detailed in Section 5.3.

Finally, prior to the real flight test, it is essential to conduct a comprehensive validation of the AFSS to guarantee the correct functioning of the system. This validation is achieved through an incremental testing approach that utilises SIL, PIL and HIL testing techniques [59]. The selection of each technique is contingent on the phase of prototyping.

- firstly, during the algorithm conception phase, the LN3 algorithms are implemented and validated on a workstation using MATLAB to ensure their correct operation. Concurrently, the software is validated following product assurance standard procedures;
- once the LN3 algorithms have been validated, the MATLAB code is translated to C/C++ and integrated with the LN2 C/C++ code. The software is then built and flashed to a BeagleBone Black board to validate both the software and processor integration. At this stage, a Speedgoat Performance Real-Time Target Machine (RTTM) is used to simulate the scenario data to ensure hard real-time conditions;
- finally, the complete AFSS prototype is validated. This includes the redundant architecture of the boards, the power distribution unit and, the data acquisition and distribution system. The objective is to validate not only the software and the processor, but also the interfaces between the processors and external systems. Scenario data

Table 7
Test-cases description.

Test-case ID	Test-case Description
NOMINAL	ELV nominal east trajectory.
DEVIATION-01	Deviation to ILL right @t = tH0+60s.
DEVIATION-02	Deviation to ILL left @t = tH0+100s.
DEVIATION-03	Deviation to VLL @t = tH0+90s.
DEVIATION-04	Initial azimuthal deviation. (225 deg.)

is further simulated using the Speedgoat Performance Real-Time Target Machine.

For each of the aforementioned phases, several scenarios are required to verify the reliability, robustness, configurability and adaptability of the developed AFSS system. It should be noted that the flight to be performed in the SALTO campaign is a 100-m vertical jump, and thus does not cover a full range of safety scenarios. To this end, the test cases maximise the range of scenarios as a combination of the following two axes.

a. Trajectory axe: test multiple real past trajectories, e.g. Ariane 5, VEGA, etc. Those trajectories may be either nominal or non-nominal, thus enabling the testing of deviation or failure scenarios (e.g. on propulsion, on separation, on actuators, etc.). Non-nominal trajectories are produced thanks to an in-house trajectory simulator [47].
b. AFSS configurability axe: test multiple AFSS configurations adapting to different navigation architectures to hybridise high/mid/low performant IMU(s) and GNSS(s), safety criteria and decision-making rules.

A subset of the test cases is presented Table 7. Fig. 6 shows the trajectories and the safety limits, i.e. the ILL and the VLL, used as input.

For the scope of this article, all the test cases outlined in Table 7 use the same AFSS configuration, where all three safety processors are configured as indicated in Table 8.

8. Results

Table 10 summarizes the results of the tests defined in Section 7, demonstrating the capacity of the AFSS to overcome non-nominal scenarios. The results reveal that all non-nominal scenarios end with a termination condition due to the violation of a safety criteria: DEVIATION-01 and DEVIATION-04 breach the Impact Predictor criterion, as would expected from the trajectory shown in Fig. 6a; while DEVIATION-02 and DEVIATION-03 exceed the VLL of the Flight Path integrity criterion, as anticipated in Fig. 6b. Note that the DEVIATION-02 trajectory would also violate the Impact Predictor criterion (see

Table 8
AFSS safety processors configuration.

	CHAIN 1	CHAIN 2	CHAIN 3
Incoming sensors	GNSS + IMU	GNSS + IMU	GNSS + IMU
Safety criteria algorithms …			
Navigation filter	EKF	IKF	EKF
Impact algorithm	FG Series	FG Series	Kepler
IIP or IIA	IIP	IIP	IIP
Decision-making …			
Max. Green time	5 s		
TERMINATE persistence counter	3 cycles		
Flight phases	(refer to Table 9)		
Processor settings …			
Cycle frequency	5 Hz		

Table 9
Flight phases configuration description.

FLIGHT PHASE ID	1	2	3	4
Description	Grounded	Near-Field Ascending	Far-Field Ascending	End of ILL applicability
Limits of applicability …				
Great-circle distance	–	–	>10 km	>1200 km
Altitude	–	–	>20 km	–
Acceleration	–	>1 g	–	–
NOT_NOMINAL condition	*consider the violation of …*			
Tracking Sources health status	X	X	X	X
Navigation Solution health status	X	X	X	X
Impact Predictor status	X	X	X	
Flight Path integrity	X	X	X	X
UNSAFE condition	*consider the violation of …*			
Tracking Sources health status	X	X	X	X
Navigation Solution health status	X	X	X	X
Impact Predictor		X	X	
Flight Path integrity		X		

Table 10
Tests results summary.

TEST ID	RESOLUTION	VIOLATION OF …
NOMINAL	CONTINUE	–
DEVIATION-01	TERMINATE	IMPACT PREDICTOR
DEVIATION-02	TERMINATE	FLIGHT PATH
DEVIATION-03	TERMINATE	FLIGHT PATH
DEVIATION-04	TERMINATE	IMPACT PREDCITOR

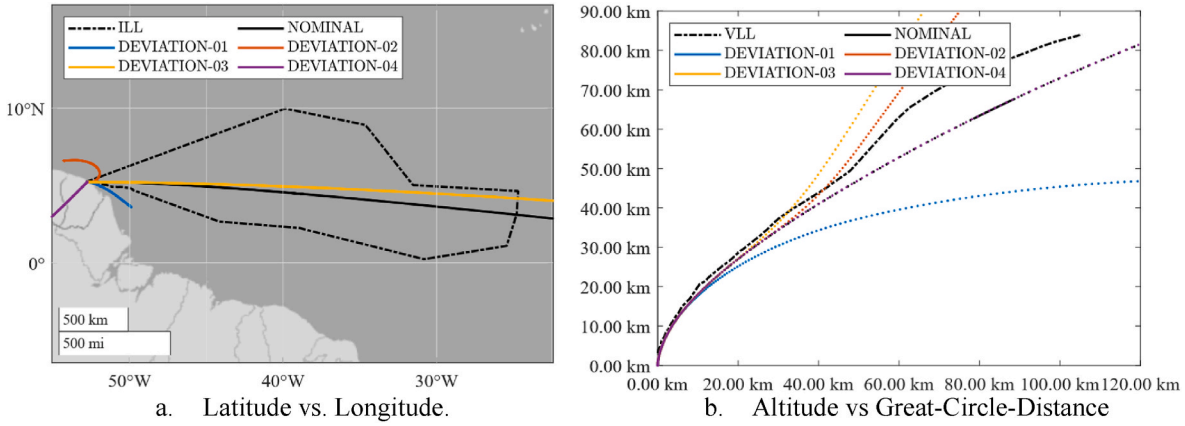


Fig. 6. Test-case trajectories. Both plots are trimmed to the safety limits range, i.e. ILL and VLL, of applicability.

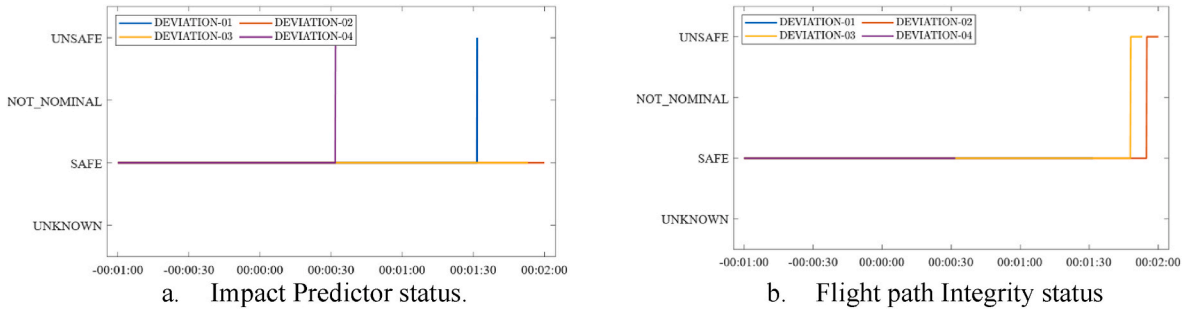


Fig. 7. Impact predictor and Flight Path integrity criteria status from the deviation test scenarios.

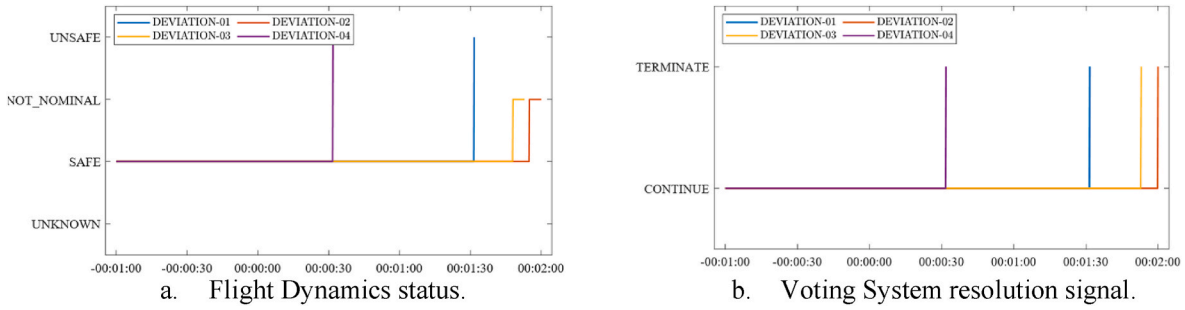


Fig. 8. Flight Dynamics status and resolution signal from the deviation test scenarios.

Fig. 6a), but the AFSS detects a violation of the Flight Path integrity criterion before the launcher exceeds the ILL (see Fig. 7). Given that all the safety chains produce identical results, only the outcomes of Safety Processor 1-Chain 1 are presented.

Furthermore, in DEVIATION-02 and DEVIATION-03 scenarios, a delay is observed between the detection of the Flight Path integrity criterion violation and the generation of the terminate signal (see Fig. 8). This result is a direct consequence of the AFSS configuration (see Tables 8 and 9). Fig. 6b shows that both trajectories exceed the VLL at a great-circle distance above the upper limit defined in flight phase 2, which considers the Flight Path violation as a condition for an UNSAFE Flight Dynamics condition (see Table 9). Therefore, the violation occurs in flight phase 3, where the Flight Path violation triggers a NOT_NOMINAL Flight Dynamics condition (see Fig. 8a), thus initialising the "green time" safety margin counter of the decision-making, set to a maximum value of 5 s, prior to the activation of the TERMINATE signal (see Fig. 8b).

Finally, the processor metrics indicate that the most demanding tasks are those related to the LN3 service, exhibiting a maximum processor load of approximately 90 % and a mean value of 40 %. This peak in processor load coincides with the non-satellized phase, during which the AFSS is required to assess the Impact Predictor criteria. Despite this workload, memory usage remains relatively low compared to the available resources. The system has 4.00 GB of flash storage, of which only 2.03 MB is utilized, and 512.00 MB of RAM, with just 2.44 MB in use. This suggests that the processing requirements leave ample room for additional tasks or future expansions.

9. Conclusions and future work

The primary challenge in developing a unified AFSS in Europe is the lack of consensus on regulations. While the U.S. regulations are the most advanced in terms of definition, they should not be applied without critical review, as their reasoning may not apply to the European CSG spaceport. A key distinction in the European approach is the use of launcher performance to define AFSS reliability metrics, rather than adhering to a fixed regulatory threshold as in the U.S. This approach enhances adaptability and can potentially lower AFSS costs for launchers with higher reliability.

The modular design approach presented in this work provides a scalable and configurable AFSS capable of adapting to diverse mission requirements, spaceports, and regulatory frameworks. The ability to select different hybrid navigation strategies and define safety criteria and decision-making rules according to flight phases enhances its configurability. The embedded interpreter plays a crucial role in improving campaign agility by reducing the need for complete software re-validation and re-qualification. Additionally, shifting hardware responsibilities to software for RAMS considerations has proven to be a strategic driver for future AFSS developments. These combined strategies contribute to an agnostic AFSS -less dependent on hardware, adaptable to various launch scenarios, and easier to update and qualify.

The proposed AFSS architecture enables fully autonomous safety functionality for ELV, and VTHL and VTVL RLV launch concepts, accommodating different operational scenarios, safety rules, and regulatory requirements by configuring the AFSS parameters in the pre-flight phase. Incorporating a critical code perspective early in the design phase is expected to reduce costs in later validation and qualification stages. For RLVs, the integration of an IVHM module enhances diagnostic capabilities, particularly during re-entry and landing, thereby improving mission safety beyond the ascent phase. This module continuously monitors key launcher subsystems, such as landing legs, grid fins, and rocket engines, ensuring safe return conditions and enabling abort decisions when necessary. Additionally, the IVHM module fosters innovation by leveraging onboard data for predictive maintenance, improving refurbishment operations for reusable vehicles.

On the hardware side, a version tailored for operational ground tests in Kiruna has been proposed. The software's agnostic design allows for seamless scalability from this test hardware to a fully functional onboard prototype, requiring only component adjustments and physical interface adaptations while maintaining the current AFSS software.

Finally, the manufactured prototype undergoes an incremental validation process using SIL, PIL and HIL testing techniques. The extensive Monte Carlo simulations, based on historical flight data, demonstrate the robustness and reliability of the system in degraded

scenarios that could represent a danger to the safety of the mission. Future developments prior to the Kiruna campaign test will focus on validating the performance of the IVHM module and corroborating its expected benefits both in the safety of the re-entry phase and in RLV refurbishment operations. With the in-flight testing milestone, the AFSS will be integrated with the Kiruna spaceport's telemetry system and the THEMIS RLV, developed by ArianeGroup SAS and CNES. This milestone represents a significant step toward demonstrating the feasibility of AFSS in European space operations, ultimately achieving a TRL 7 maturity level and delivering the first fully autonomous and operational AFSS in Europe, validated under real flight conditions.

CRediT authorship contribution statement

Alejandro Sabán-Fosch: Writing – original draft, Validation, Investigation, Conceptualization, Software, Funding acquisition, Visualization, Methodology, Formal analysis. **Eduard Diez-Lledó:** Writing – review & editing, Funding acquisition, Supervision, Conceptualization, Project administration. **Manel Soria:** Writing – review & editing, Supervision. **Miquel Sureda:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alejandro Sabán-Fosch reports financial support was provided by Generalitat de Catalunya Ministry of Research and Universities. Eduard Diez-Lledó reports financial support was provided by European Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

I would like to express my gratitude to my colleagues, *Pablo Miralles, Jordi Martin, Marc Aubach, Carles Pou, David Vallverdú* and *Nil Martin*, for their valuable input and suggestions. Lastly, this work would not have been possible without the generous financial support of the *European Commission [grant number 101082007]*; and the *Generalitat de Catalunya Ministry of Research and Universities [grant number 2022 DI 077]*.

References

[1] The European Space Agency (ESA), Technology Readiness Levels (TRL). https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Shaping_the_Future/Technology_Readiness_Levels_TRL. *April 2024*.
[2] Centre National d'Études Spatiale, [Quézako?] Un système de localisation infaillible à bord des fusées, CNES, https://spacegate.cnes.fr/fr/quezako-un-systeme-de-localisation-infaillible-bord-des-fusees, 24 August 2023. *April 2024*.
[3] Centre National d'Études Spatiale, [Lanceurs] Kassav 2, sur la piste d'une sauvegarde automatisée, CNES, https://cnes.fr/fr/lanceurs-kassav-2-sur-la-piste-dune-sauvegarde-automatisee, 19 April 2021. *April 2024*.
[4] GTD System & Software Engineering, KASSAV2: GTD and SAFRAN to develop a new autonomous system for Space Vehicle's Flight Safety. https://www.gtd.eu/es/noticias-y-eventos/kassav2-gtd-and-safran-develop-new-autonomous-system-space-vehicles-flight-safety, 2021. *April 2024*.
[5] GTD System & Software Engineering, Andøya's GTD launch control System breaks new ground, enabling Europe's first Continental orbital launch. https://www.gtd.eu/en/news-and-events/andoya-s-gtd-launch-control-system-breaks-new-ground-enabling-europes-first, 2023. *April 2024*.
[6] Sener Aerospacial, SENER Aeroespacial and Gilmour Space to develop Autonomous Flight Termination System for Eris launch vehicle. https://www.group.sener/en/noticias/sener-aeroespacial-and-gilmour-space-to-develop-autonomous-flight-termination-system-for-eris-launch-vehicle/. April 2024.
[7] European Commision (24 October, Cordis - EU research results - smart Avionics for Flight tErmination SysTems, SAFEST (2022). https://cordis.europa.eu/project/id/101082662. *April 2024*.
[8] Sergio Ramírez , Maria Rodríguez , Alba Rozas , Jesus Zurera, Mariano Sanchez, European autonomous flight termination systems based in smart avionics, in: Proceedings of the 75th International Astronautical Congress, IAC. Milan, Italy, 2024 doi: 10.52202/078373-0042.
[9] Marc Schwarzbach, Daniel Ridley, Christian Schmierer, Sven Weikert, Julia Gente, Stephan Schuster, Derivation of Requirements for Autonomous Flight Termination System FTS next, June 2022.
[10] Sven Weikert, Julia Gente, Marc Schwarzbach, Daniel Ridley, Christian Schmierer, Stephan Schuster, Next Gneration Flight Termination System for Launchers, 2022.
[11] Astos Solutions GmbH, Ftsnext - next Generation Flight Termination Systems for Launchers - Kick-Off Presentation for Final Review, 2023.
[12] James Dean, Only on Falcon 9: automated system can terminate SpaceX rocket launches. https://eu.floridatoday.com/story/tech/science/space/2017/03/11/spacex-autonomous-flight-safety-system-afss-kennedy-space-center-florida-falcon9-rocket-air-force-military/98539952/, 2017. *April 2024*.
[13] Chris Gebhardt, Air Force reveals plan for up to 48 launches per year from Cape Canaveral. https://www.nasaspaceflight.com/2017/03/air-force-reveals-48-launches-year-cape/, 2017. *April 2024*.
[14] Rocket Lab ISA, Inc. (n.d). Rocket lab debuts fully autonomous flight termination System. https://www.rocketlabusa.com/updates/rocket-lab-debuts-fully-autonomous-flight-termination-system/. Retrieved April 2024.
[15] Jeremy Egger, New NASA safety System enables rocket lab launch from wallops. https://www.nasa.gov/centers-and-facilities/wallops/new-nasa-safety-system-enables-rocket-lab-launch-from-wallops/, 2023. *April 2024*.
[16] Bob Ferrell, Sam Haley, Autonomous system for launch vehicle range safety, AIP Conf. Proc. 552 (1) (2001) 686–692, https://doi.org/10.1063/1.1357994.
[17] James B. Bull, Raymond J. Lanzi, An Autonomous Flight Safety System, 2008.
[18] Jamie Adkins, NASA releases autonomous flight termination unit software to industry. https://www.nasa.gov/centers-and-facilities/wallops/nasa-releases-autonomous-flight-termination-unit-software-to-industry/, 2022. *April 2024*.
[19] Takafumi Akiyama, Toru Ishikawa, Naomi Obuchi, Naomasa Dairaku, Akito Ofuchi, Aya Asamura, Hidenori Hasegawa, Space Engineering Development Co, Development of autonomous flight termination software. Proceedings of the 75th International Astronautical Congress, IAC. Milan, Italy, October 2024, https://doi.org/10.52202/078377-0008. , Ltd.(SED), and JAXA.
[20] Aya Asamura, Hidenori Hasegawa, Shin'ichi Goto, Yuto Takase, Takuya Yamada, Karebu Suzuki, Ryo Kato, Yuichiro Kaneko, Kazuhiko Ohkawa, Hotaka Nakao, Atsushi Urayama, Koji Sunami Jaxa, Mitsubishi Precision Co.,Ltd, Japan's first in-flight experimentation of autonomous flight termination software using a sounding rocket, in: Proceedings of the 75th International Astronautical Congress, IAC. Milan, Italy, October 2024, https://doi.org/10.52202/078373-0048.
[21] Philippe Baptiste. CNES, V de travail au 15 juillet 2024. Arrêté Portant Réglementation De L'Exploitation Des Installations Du Centre Spatial Guyanais, 2024.
[22] République française, LOI N° 2008-518 Du 3 Juin 2008 Relative Aux Opérations Spatiales, 2008.
[23] Yannick d'Escatha. CNES, Arrête Portant Réglementation De L'Exploitation Des Installations Du Centre Spatial Guyanais, 9 December 2010.
[24] République française, Arrêté Du 31 Mars 2011 Relatif À la Réglementation Technique En Application Du Décret N° 2009-643 Du 9juin 2009 Relatif Aux Autorisations Délivrées En Application De la Loi N° 2008-518 Du 3 Juin 2008 Relativeaux Opérations Spatiales, 31 May 2011.
[25] Peter Lindström, Makro Kohberg, Mats Tyni, Lennart Poromaa, Swedish space Corporation, Esrange Safety Manual (2020).
[26] Range Safety Group. Range Commanders Council, Flight Termination Systems Commonality Standard, 2019, p. 319, 19.
[27] Range Safety Group. Range Commanders Council, Global Positioning and Inertial Measurements Range Safety Tracking Systems Commonality Standard, 2011, p. 324, 11.
[28] Federal aviation administration, 14 CFR PART 417 - Launch Safety, 2022.
[29] Federal Aviation Administration, 14 CFR PART 450 - Launch and Reentry Licence Requirements, 2022.
[30] Federal Aviation Administration, Flight Safety Analysis Handbook - Version 1.0, 2011.
[31] Civil Aviation Authority, Guidance for Launch Operator and Return Operator Licence Applicants and Licensees, 2021.
[32] Patsy Reddy, Governor-General, Outer Space and high-altitude Activities (Licences and Permits) Regulations, 21 August 2017, 2017.
[33] Australian Space Agency, Flight Safety Code, August 2019.
[34] Radio Technical Commision for Aeronatics, Software Considerations in Airborne Systems and Equipment Certification, DO-178C, 2011.
[35] ECSS Secretariat ESA-ESTEC, ECSS-E-ST-10-06C Space Engineering – Technical Reqirements Specifcation, 6 March 2009.
[36] ECSS Secretariat ESA-ESTEC, ECSS-E-ST-10C Rev.1 - System Engineering General Requirements, 2017.
[37] ECSS Secretariat ESA-ESTEC, ECSS-E-ST-40C Space Engineering – Software (2009).
[38] ECSS Secretariat ESA-ESTEC, ECSS-Q-ST-30C-Rev, 2017, 1 - Dependability.
[39] ECSS Secretariat ESA-ESTEC, ECSS-Q-ST-30-02C - Failure Modes, Effects (And Criticality) Analysis (FMEA-FMECA), 2009.
[40] ECSS Secretariat ESA-ESTEC, ECSS-Q-ST-40C-Rev.1 - Safety, 2017.
[41] ECSS Secretariat ESA-ESTEC, ECSS-Q-ST-80C Rev.1 Space Product Assurance – Software Product Assurance, 2017.
[42] Pascal Roques, MBSE with the ARCADIA method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), 2016. Toulouse, France. hal-01258014.
[43] Charlie Höglund, Guidelines for UML or SysML modelling within an enterprise architecture. https://www.diva-portal.org/smash/get/diva2:1107088/FULLTEXT01.pdf, 8 June 2017. *April 2024*.
[44] Nesrine Badache, Pascal Roques, Capella to SysML Bridge: a Tooled-up methodology for MBSE interoperability. 9th European Congress on Embedded Real

Time Software and Systems (ERTS 2018), TOULOUSE, France, January 2018 hal-01710529.
[45] C. Hourtolle (CNES), RNC-CNES-Q-HB-80-507: Analyse De Surete De Fonctionnement Du Logiciel Par La Methode De L'Arbre De Causes, 2 June 2008. Version 3.
[46] David Vallverdú, Carles Pou, Mariona Badenas, Eduard Diez, Application of a Hybrid Navigation System for an Autonomous Space Air-Launched Vehicle (ERTS 2018), TOULOUSE, France, January 2018 hal-02156070.
[47] Alejandro Sabán, Jordi Martin, Nil Martin, Eduard Díez, Design and Validation tool for modular flight software in the domain of newspace launch services development, in: 2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions & Engineering (FAR), Heilbronn, Germany, June 2022.
[48] Eric A. Wan, Rudolph van der Menve, The unscented kalman filter for nonlinear estimation. Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), 2000, pp. 153–158, https://doi.org/10.1109/ASSPCC.2000.882463.
[49] Carles Pou, David Vallverdú, Eduard Diez, Design of an On-board flight safety system for space microlaunch vehicles, in: 8th European Conference for Aeronautics and Space Sciences (EUCASS), January 2018, https://doi.org/10.13009/EUCASS2019-657.
[50] Jaemyung Ahn, Woong-Rae Roh, Noniterative instantaneous impact point prediction Algorithm for launch operations, J. Guid. Control Dynam. (March-April 2012), https://doi.org/10.2514/1.56395.
[51] Montenbruck Oliver, Markus Markgraf, Wolfgang Jung, Barton Bull, Wolfgang Engler, GPS based prediction of the instantaneous impact point for sounding rockets, Aero. Sci. Technol. 6 (4) (2002) 283–294, https://doi.org/10.1016/S1270-9638(02)01163-X.
[52] Kai Hormann, Alexander Agathos, The point in polygon problem for arbitrary polygons, Comput. Geom. 20 (3) (March 2021) 131–144, https://doi.org/10.1016/S0925-7721(01)00012-8.
[53] Alan Murta, GeneralPolygonClipper (Version 2.32). https://github.com/rickbrew/GeneralPolygonClipper, August 2009.
[54] Nil Martín, Carla Navarro, Eduard Díez, Alejandro Sabán, Jordi Martín, Magda Escorsa, Missionisable on-board flight safety based on real-time autonomous decision-making for microlauncher services, in: 9th European Conference for Aeronautics and Space Sciences (EUCASS), 2022, https://doi.org/10.13009/EUCASS2022-4841.
[55] RTEMS org, What is RTMES?. https://www.rtems.org. *April 2024*.
[56] DENX Software Engineering, U-Boot (Version 2024.04). https://source.denx.de/u-boot/u-boot, 2 April 2024.
[57] BeagleBoard Org, Beaglebone Black Release 1.0.20240415, 2024.
[58] Yulin He, The Motor Industry Software Reliability Association, MISRA), MISRA C++:2008, in: Guidelines for the Use of the C++ Language in Critical Systems, 2010, 978-1-906400-04-0 PDF.
[59] T. Erkkinen, M. Conrad, Verification, Validation, and Test with Model-based Design, SAE Technical Paper, 2008, https://doi.org/10.4271/2008-01-2709, 2008-01-2709.