Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

IAC-24-D2.5.9

European Autonomous Flight Termination Systems Based in Smart Avionics

S. Ramírez*[a], M. Rodríguez[b], A. Rozas[c], J. Zurera[d], M. Sanchez[e]

[a] *SENER Aeroespacial, Severo Ochoa, 4, 28760 Tres Cantos, Madrid, Spain,* sergio.ramirez@aeroespacial.sener
[b] *SENER Aeroespacial, Severo Ochoa, 4, 28760 Tres Cantos, Madrid, Spain,* maria.rodriguez@aeroespacial.sener
[c] *SENER Aeroespacial, Severo Ochoa, 4, 28760 Tres Cantos, Madrid, Spain,* alba.rozas@aeroespacial.sener
[d] *SENER Aeroespacial, Severo Ochoa, 4, 28760 Tres Cantos, Madrid, Spain,* jesus.zurera@aeroespacial.sener
[e] *SENER Aeroespacial, Severo Ochoa, 4, 28760 Tres Cantos, Madrid, Spain,* mariano.sanchez@aeroespacial.sener
* Corresponding Author

Abstract

Operating a launcher is not exempt from risks. Even if we are talking about a reliable and tested system, the launchers need to operate in a very harsh environment. Thus, any issue can cause a failure and the loss of the control on the system. During the ascent phase, any failure implies risks to human lives since it would be a huge uncontrolled vehicle full of flammable fuel. Therefore, the launchers are tracked and monitored during the ascent phase, and in case any issue is detected, the launcher needs to immediately terminate (typically exploding).

A traditional flight termination architecture ensures independency from the vehicle functional chain using a radar network with human involvement in the decision-making process. This means: (i) considerable budget share for infrastructure and operations, (ii) limited flexibility (radar network needed), (iii) vehicle monitoring restricted to LOS conditions and (iv) delay inherent to communications and human reaction.

The autonomous flight termination systems (AFTS) determine the safety of the flight by processing different tracking inputs and comparing current estimated state to flight rules (also known as mission rules), defined by the user during flight missionization phase. By processing the mission rules directly on-board, the reaction time is reduced, and the telemetry downlink is no longer required.

Within Europe, there is no clear standard on the design nor operation of an AFTS. The critical part of this type of standards is related to managing the idiosyncrasy of the flight regulation in each country, making it difficult to have a common standard in Europe. The solution proposed is to have a highly configurable unit, in which the range safety officer could even include proprietary software for the termination logic, thus adapting to each specific local flight regulation without the need to perform any factory customization.

The paper describes the general problem and the proposed solution for a European Autonomous Flight Termination System highly configurable by the user, which make is suitable for a broad range of launchers and countries. Sener is developing an AFTU demonstrator in the frame of the RD EC Horizon Europe programme.

Keywords: launchers, flight safety, flight termination, autonomous, termination logic, fail-safe

Acronyms/Abbreviations

AFTS	Autonomous Flight Termination System
AFTU	Autonomous Flight Termination Unit
cFS	Core Flight Software
DKE	Dynamic and Kinematic Environment
EGSE	Electrical Ground Support Equipment
EM	Engineering model
EoR	End of Responsibility
FTS	Flight Termination System
GNC	Guidance Navigation and Control
GNSS	Global Navigation Satellite System
IIP	Instantaneous Impact Point
IMU	Inertial Measurement Unit
LoCRAFTS	Low-Cost Robust AFTS
MGSE	Mechanical Ground Support Equipment
MIA	Modular Integrated Avionics sMart Integrated Avionics
MPSoC	Multi Processor System-on-Chip
MR	Mission Rules
OSAL	Operating System Abstraction Layer
PSP	Platform Support Package
SAFEST	Smart Avionics for Flight TErmination SysTem
SC	Safe Condition
SOC	System on Chip
SSLA	SENER Service Layer API
SWaP	Size, Weight and Power
TCU	Termination Command Unit
TSP	Time and Space Partitioning

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

1. Introduction

The traditional flight termination approach involves a complex interaction between the launcher, the ground systems and their operators (see Fig.1). This interaction brings a dependence on the availability of ground infrastructures that are directly related to the start cadence can have an impact. Also intervening humans as a potential source of delays (typically 3-5 seconds reaction time) in intervention during flight. The inclusion of the ground segment in the safety chain of the launcher flight also has a direct impact on the cost since their use implies an important fee.

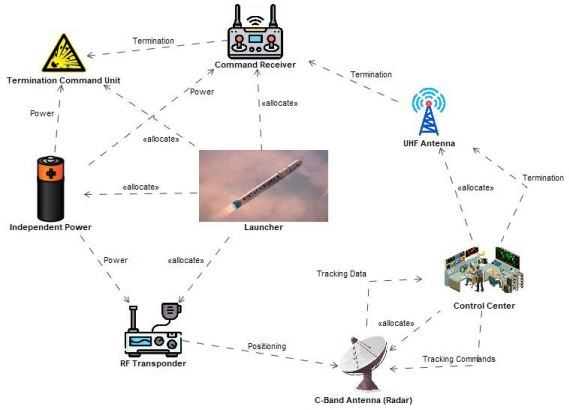

Fig. 1. Traditional FTS operations schematic

An Autonomous Flight Termination System (AFTS) consists in the entire logic behind the decision-making (safety systems, ground infrastructure and human decision-maker) as an unmanned system on-board the launcher, as depicted in the following figure. This can increase the launch cadence, since several launchers can be operated practically at the same time, and independently of the supporting aviation safety infrastructure on the ground.

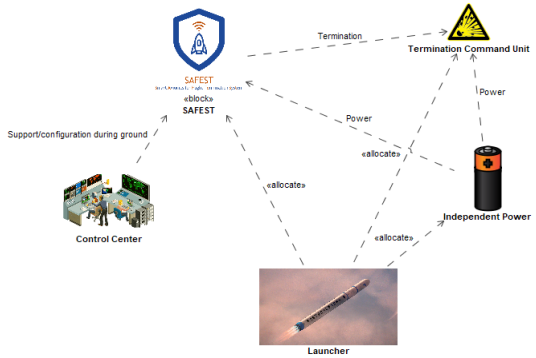

Fig. 2. AFTS operations schematic

The AFTS is an independent, redundant system on board the launcher, which must function despite a failure of the carrier avionics. The system is completely decoupled from functional avionics and has its own positioning system, sensors and power sources. Accordingly, it can be developed as an independent system and used on a wide variety of launchers. The AFTU provides the termination signal to the actuator, whereby the flight termination can be achieved either via thrust scheduling or targeted explosion.

Following table shows a comparison between the different elements necessary to operate the different types of termination systems.

Table 1. AFTS vs FTS elements comparison

Traditional FTS	AFTS
Flight systems	
HW Unit	
Safe & Arm	
Ordnance	
Receiver	-
FTS logic box	-
Battery	-
UHF antenna	-
Hybrid coupler	-
Metric Tracking sources (RCC 324)	
GPS	
L-band antenna	
Couplers	
Power distribution box	
Vehicle battery	
Telemetry encoder	-
Telemetry transmitter	-
S-band antenna	-
-	IMU/INS
-	Flight computer
-	External tracking inputs
Radar transponder	
Transponder	-
C-band antenna	-
Hybrid coupler	-
Power distribution box	-
Vehicle battery	-
Ground systems	
Command transmitters	
Power supplies	-
Antennas	-
Amplifiers	-
Telemetry receivers	
Antenna	-
Decoders	-
Ground comm networks	-
Radars	
Radar sites	-
Ground comm networks	-
Timing infrastructure	-
Mission flight control	
MFCO	-
Telemetry officer	-
Certified display	-
Others	
Preflight testing	

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

1.1 Existing Solutions

In the US, NASA has developed its own AFTU SW known as CASS. This SW consists in a set of generic functionalities implemented in C++, which are used to customize and missionize each launch. Currently, only two American companies are using AFTS system, developed by themselves, SpaceX and Rocket Lab.

The FAA will require AFTS for all the launches from American soil by 2025. So far, no launch from European soil have been performed with AFTU.

In Europe, there is only known one project called Kassav-2, led by CNES, which does not seem to be still continued.

Since the AFTU units are not widely used or even developed, there is a lack of standards to be followed for its design in Europe. From the US, there is a Standard called RCC-319 [1], focused on Flight Termination Systems for launchers that includes several chapters for the design and testing of AFTU.

1.2 Market Needs

Traditional termination systems are costly and complex, and also, they jeopardize the increase of the launch cadence. Introducing AFTS will provide following benefits to the launcher providers:

- Decrease of recurrent price per launch, due to the removal of complex equipment and ground stations.
- Increase of cadence since no dedicate ground stations and officer are required.
- Increase of operations, avoiding limitations from the use of ground stations (e.g. line-of-sight constraints).

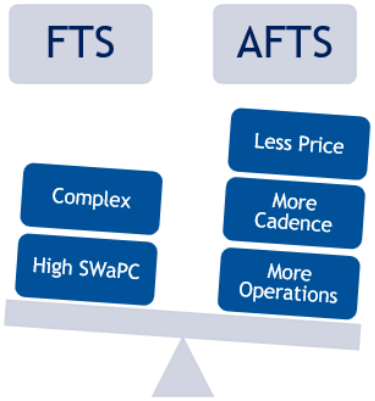

Fig. 3. AFTS main advantages schematic

To be competitive, the European launch sector has to main objectives, (i) lowering the launch recurrent prices, and (ii) succeeding reusability. Operating AFTS is a key element to achieve the first objective, and partly the second one.

In order to design a AFTS system, following main requirements should be taken into account, which have been extracted from relevant actors in the European sector (e.g. launch providers, range officers, space agencies):

- Drastic reduction of SWaP and cost of the on-board equipment.
- High-reliability, mainly achieved by HW design avoiding any single point failure (including protection against external interferences), and by the maximum SW category (i.e. SW Cat. A from ECSS).
- Compatibility with current traditional flight termination systems, since these units would not be operated till fully certified, including dedicated flight campaigns.
- Availability to customize the termination logic, which the possibility to include own proprietary SW, which may not be possible to be shared with the AFTS manufacturer due to export control restrictions.
- Scalability to different launch sizes or needs. The AFTS solution should be designed for small launchers, with the capability to be extended to greater launch sizes.

1.3 SAFEST Project

SAFEST is proposing an Autonomous Flight Termination System design and architecture that is highly customizable by the user, to cover different countries regulations, and to adapt to different launcher types, without the need of a new development.

In particular, the project is focused on the development of a AFTU unit, the equipment that executes the termination logic taking the data from the tracking inputs. The objective of the project is to reach a TRL 5-6 for the AFTU SW and the avionics.

Fig. 4. SAFEST project consortium

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

2. AFTU Design

The architecture of SAFEST unit is described in Fig 5. Each SAFEST unit will contain an IMU and a GNSS receiver and the data coming from these sensors will be blend in the data fusion module to obtain a robust navigation solution. The GNSS measurements will be fused in a loosely-couple scheme of an Extended Kalman Filter, see [2].

This solution will be provided to the mission rules so that they can check the violation of the mission rules. Note that the mission rules will also receive data from external tracking inputs and redundant unit (cross-strap interface) to perform the check of the mission rules violation. Note that inside the SAFEST unit it is also envisaged the implementation of a fail-safe system in charge of monitoring the power supply to detect power failures and generate the termination command.

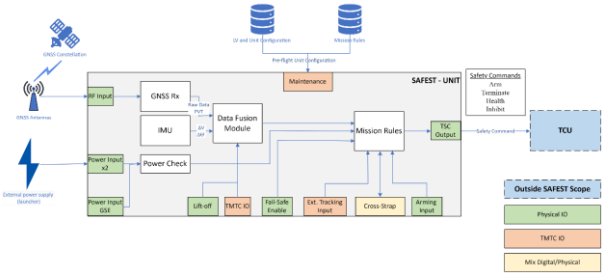

Fig. 5. SAFEST unit architecture

Following table provides a more detailed description of key features of the AFTU unit design proposed

Table 2. SAFEST unit design key features

Feature	Description
Cross-Strapped Interface	To increase the reliability of the AFTS, SAFEST unit can operate in cross-strap redundancy. The following information will be transmitted between the units: • Fail-safe inhibiting signal: This logical signal indicates that the fail-safe system is enabled and operating adequately. The receiver of this signal will inhibit the generation of the termination command since in case of unit failure the redundant unit could still operate. • Health Status: The health status is a digital signal describing the status of the mission rules checks. The receiver shall check if the health status is valid (no SW failure, valid tracking inputs available…) and inhibit the start of green time algorithm, in case of loss of all valid tracking inputs.
Fail-Safe System	The Fail-Safe system monitors the power input, and generates the termination command in case a power failure is detected. The Fail-Safe system can be enabled/disabled by the user and is inhibited in the scenario where the Fail-Safe system of the redundant unit is enabled and operating correctly.
Mission Rules Config.	The behaviour of the mission rules can be configured by the user to its full extent by means of a configuration file. Note that the mission rules SW will be provided to the user in an open format, therefore the user can configure any new mission rule.
Integrated Navigation Solution	The baseline unit includes a low-cost hybrid navigation solution that provides robust estimation of the position and velocity of the vehicle at high frequency. Besides, it provides accurate acceleration and angular velocity estimations, necessary to detect faulty events (e.g. vehicle tumbling).
External Tracking Inputs	The mission rules can manage up to three external tracking inputs provided by the user. Three types of interfaces are envisaged: INS, GNSS and hybrid interface.

2.1 Operations

The behaviour of the unit can be fully configured by the user by means of several inputs:

- Master Arm: The master arm enables/disables the power required to generate the terminate command. Therefore, if the master arm is not enabled the terminate command can't be generated.
- Logic Arm: The logic arm enables/disables the execution of the mission rules SW. The logic arm should be enabled during the flight and can also be enabled before flight to perform tests.
- Liftoff indicator: Indicates to the unit that the liftoff has already occurred.
- Fail-Safe system enabling: The fail-safe system can be enabled or disabled by means of a physical input. The user can disable the fail-safe system to avoid undesired generation of the termination command.

There are two main phases involved in the operation of an AFTU, (i) missionization and (ii) flight. The processes associated for these phases are described at high-level below.

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

Configuration & testing

The mission rules should be configured for each flight depending on the launcher, the target orbit and the launchpad. To increase the versatility of the Mission Rules SW the user can configure the mission rules by means of a defined configuration file, or even include their own mission rules logic.

Also, the testing of the unit is possible thanks to the dedicated HWIL interfaces.

Flight

On ground, the unit should be powered on using the GSE power input since the unit has been designed to avoid termination when this input is detected. Note that also the battery power should be included to transition from the GSE power to the battery power that will be used during the flight. For safety purposes it is recommended to enable the fail-safe system after the transition from the GSE power to the battery power. The logic arm should be enabled by the user prior to the master arm enabling to check that the behaviour of the mission rules is adequate (no termination generation and valid tracking inputs).

Once the transition has been performed the user shall enable the master arm and the fail-safe system for the liftoff. It is recommended to define an eventual safe condition in the form of time from liftoff lower than a threshold or altitude lower than a threshold to avoid termination close to the launchpad.

During the flight the mission rules will be checked, unless an eventual safe condition is met, until the permanent safe condition is met. After the permanent safe condition is met, the termination command will not be generated even if a power failure is detected or a mission rule violation. At this point the user can power off the unit without the risk of termination command generation.

If the End of Responsibility mode has not been reached but the user wants to power off the unit, it shall disable the master arm and fail-safe system first and then power off the unit to avoid the generation of the termination command by the fail-safe system.

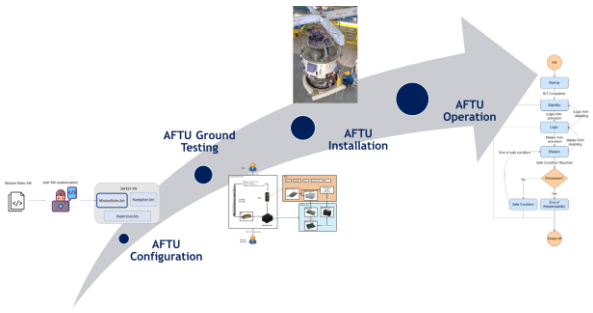

Fig. 6. SAFEST unit configuration and operation

2.2. Avionics

The AFTU system that is being developed in the SAFEST project, mainly consists in a series of algorithms for detecting the need of a flight termination. These algorithms are designed and modeled in a high-level mathematical simulation and modeling environment, but ultimately coded in C language for real-time execution. Thus, the AFTU product comprises several hardware and software subsystems that form the avionics of the system, namely:

- An electronic box with power supply adaptation, backup batteries and several analog and digital input/output signals to communicate with the rest of the launcher. The AFTU system must be powered from the launcher main power line, but, given the criticality level of its operation, it must be able to keep functioning in the event of a power loss for at least a specified time by means of its backup batteries. This electronic system must have an output to command the termination of the launcher flight. The actual details of the termination procedure are up to the launcher, and the AFTU system just needs to be compliant with the termination signal interface of the specific launcher it is mounted on.
- A processing board in which the AFTU software is executed. The AFTU software not only comprises the termination algorithms (called Mission Rules and described in the following section), but also an avionics software stack with several auxiliary FPGA and software services. This processing board is based on the MIA Execution Platform, that is designed and maintained by SENER and is described in the next subsection.
- A set of navigation sensors (an IMU and GNSS receiver). These sensors are required so that the AFTU system can independently infer the current position, velocity and dynamics of the launcher, which are a necessary input for the Mission Rules algorithms. In addition, the AFTU system may admit the connection of auxiliary external navigation sensors, e.g. some of the ones from the launcher navigation system itself. In this case the AFTU may implement a configurable weighting or logic algorithm to combine the inputs of the different sensors.

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

2.3. MIA Execution Platform

The core part of the avionics in the AFTU is the MIA Execution Platform where the termination algorithms are run. MIA is a generic execution platform for space and flight related applications whose architecture is designed by SENER [3]. The main idea behind MIA is having a set of already available and tested common services and features for the necessities of flight and space applications, which can be reused from mission to mission. This way, when using MIA in a given project, the developers can focus only on implementing the specific mission-related functionality, which is deployed as an application or set of applications on top of the MIA platform. This has a huge effect in reducing the time and cost of the software development and testing activities in space-related projects.

MIA is composed by several software modules deployed on top of a hardware processor. The full potential of the MIA platform is achieved when this hardware processor is of the Multi Processor System-on-Chip (MPSoC) type. In this case the processor includes both programmable FPGA fabric as well as several processing cores. This allows the developer to allocate the mission functionalities either as FPGA blocks (for fast time-constrained low-level operations) or as traditional software to be executed in the cores (for the more complex algorithms and mode management of the unit, etc.). In any case, the MIA platform can be ported to simpler processing hardware and it is not constrained to a specific architecture or manufacturer.

MIA has a layered architecture that enhances the portability and reusability of its components. The main layers are from bottom to top:

- Hardware layer: it consists of the actual physical components in which the upper software layers are executed, including the FPGA fabric, the processing cores, peripherals, memory devices and interconnect logic.
- Time and Space Partitioning (TSP) layer: if present, this layer provides the ability to split the missions' functionalities into different partitions guaranteeing independence between them. The layer's functionality is based on a hypervisor that spatially or temporally allocates the hardware resources to each partition.
- Operating System (OS) layer: it supports several operating systems or a bare-metal configuration if no OS is present.
- Service Layer: it provides generic flight software services and functionalities to the applications, as well as providing a standardized interface for application development. This layer is based on NASA's core Flight System (cFS) [4], and specifically its core Flight Executive (cFE) with its common services and modules for messaging, TM/TC, timing and scheduling, etc. The SENER Service Layer API (SSLA) acts as the upper interface of this layer, enhancing the abstraction to the rest of the platform and wrapping the underlying services and functions for the applications. SSLA also acts as a standard interface for the software applications to communicate with the FPGA blocks if present.
- Application Layer: it contains the software applications that provide the mission-specific functionalities.

All the software components that make up MIA communicate with each other through standardized interfaces for modularity. This standardization and abstraction allow the designer or developer to tailor the individual components for the specific mission or project in which MIA is being used. In fact, the software component of any of the layers may be supplied by an external company or be open-source in nature. Also, any of the layers may be removed in a specific mission configuration (e.g. a bare-metal software can be implemented in MIA without using any operating system).

In the case of the SAFEST AFTU the selected MIA configuration is depicted in Fig. 7. As can be seen, the hardware layer is fulfilled by the Xilinx Zynq-7020 MPSoC which includes two ARM processing cores and some FPGA fabric. In turn, the TSP layer consists in the XtratuM XNG Hypervisor by FentISS [5], in which two separate partitions are deployed. One of them is destined for executing the Navigation application that reads the IMU and GNSS sensors and provides a navigation solution in real-time. This solution is fed to the other partition, where the Mission Rules application is executed.

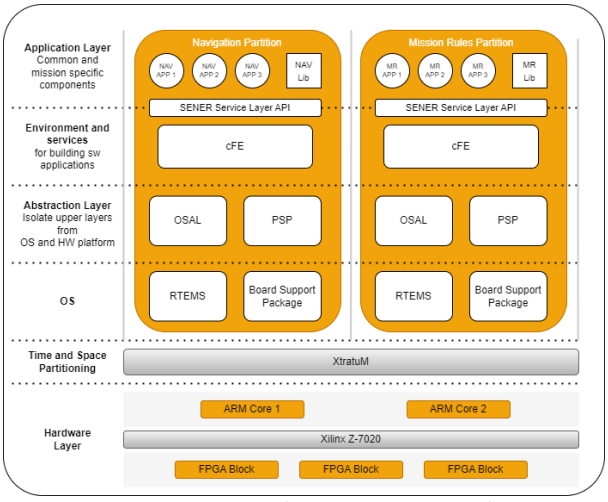

Fig. 7. Execution Platform architecture of the AFTU

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

Both partitions use a RTEMS version 6 operating system, with the specific Board Support Package (BSP) for the XNG hypervisor. RTEMS and the BSP are pre-qualified for ECSS compliance at a criticality level B by Embedded Brains, GmbH [6]. The Operating System Abstraction Layer (OSAL) and Platform Support Package (PSP) layers with which CFE is adapted to the underlying OS and hardware are tailored for this configuration.

The service layer of both partitions contains the cFE and SSLA combination, as well as a series of hardware and software services. Finally, the application layer of each partition contains a set of SSLA-compliant applications for implementing the Navigation and Mission Rules algorithms respectively.

The FPGA/SW allocation of functionalities has been done aiming to have the FPGA dedicated to managing the low-level access to hardware devices and external sensors. In particular, the FPGA is tasked with:

- acquiring the navigation sensors (IMU and GNSS)
- maintaining the system's on-board time
- outputting telemetry
- managing the interpartition communication of the navigation solution from the Navigation to the Mission Rules partition.

This allows the AFTU to have a lighter, less stressed and easier to test software which is also less dependent on external manufacturers. The software is therefore dedicated to:

- executing the high-level navigation and mission rules algorithms
- maintaining the mission modes
- ensuring the correct order of execution and timing of the different tasks and applications
- detecting and reporting failure situations

The software architecture of the applications is carefully designed for robustness, real-time compliance and reusability.

2.4 Mission Rules

The mission rules (MR) are a set of rules used to evaluate on-board the safety of current flight, defined by the user prior to the takeoff. The algorithms implementation is based on processing tracking input information and cross-checking the difference between the stored nominal trajectory and the actual trajectory followed by the launcher.

The state vector is computed and updated using the information coming from the navigation solution (IMU, GNSS or hybrid IMU-GNSS). Variables related to the vehicle are computed (position, velocity, ground track, azimuth, …) and, subsequently, used to define the MR conditions. An example of unsafe flight is if the azimuth is not within the predefined range.

The MR algorithms perform a wide range of checks:

- boundary check
- table check
- gate check
- moving-gate check
- user defined checks
- green time
- tumbling

After executing the MR, a terminate or a safe condition (SC) signal can be triggered as output. The termination flag means that some MR is not being fulfilled, while a safe condition indicates a safe flight or that some conditions are being violated but it does not pose any risk. The safe condition is categorized as temporal, if the MR block is executed again, or permanent, also referred to as end of responsibility (EoR), when the MR are not executed again until a unit reset.

For example, the termination flag would rise if the ground track or IIP of the launcher fall out of the allowed flight area. In contrast, the SC flag and even the EoR are activated if the launcher leaves the allowed region crossing a gate that guarantee the safety of the flight.

To recapitulate the process followed for MR is depicted in Fig. 8, consisting in a series of conditions defined by the user, which are later used for termination or safe purposes.

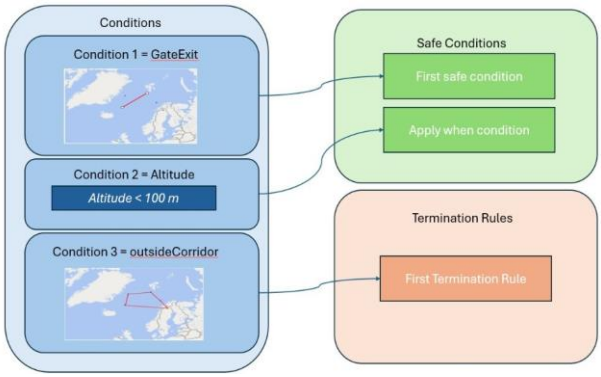

Fig. 8. Mission Rules logic

The mission rules SW is characterized by its highly configurability, the user can define its own conditions, and the unit can be used in launches from different launch sites.

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

2.5 Prototyping

The Breadboard Model of SAFEST is based on the Zybo Z7 (Z7-20) development board. The Zybo Z7 is a digital circuit development board built around the Zynq-7000 SoC, which integrates a dual-core ARM Cortex-A9 processor and a Xilinx 7-series FPGA on a single chip. To this SoC, the Zybo Z7 adds all the peripherals needed to operate as a single-board computer: among others, DDR3L memory device, Quad-SPI FLash memory device, USB, and Ethernet interfaces and 6 Pmod ports.

This development board will be connected to:

- Engineering models of the selected sensors (IMU and GNSS) to gather and process real measurements.
- External computers simulating the unit commands and gathering the outputs.

The Pmod ports can be used to implement SPI, UART or CAN communication protocols, which offers great flexibility in establishing communication between the SAFEST Execution Platform and sensors or between SAFEST and external computers.

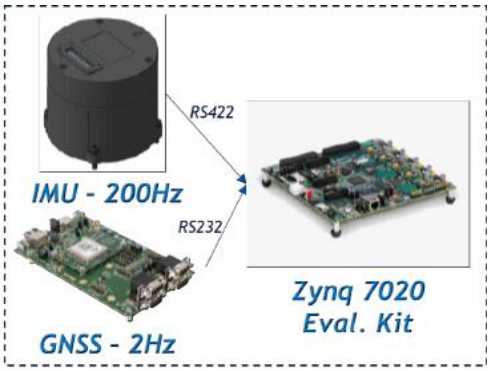

Fig. 9. Mission Rules logic

3. Results

A validation process has been performed to accomplish that the implemented logic meets its purpose and visualize the results.

The methodology selected has consisted of creating several scenarios to test the mission rules conditions and outputs developed in Model-in-the-loop (MIL), receiving data from DKE to simulate an idealized environment. If the results shown are as expected, it is ensured that the simulator works properly. The procedure continues autocodifing the Matlab/Simulink functions, to translate them to C-code. A tool called S-function builder is used and the MISRA guidelines followed to ensure reliability, readability and testability of the source code. Like this, the functions are tested at Software-in-the-loop (SIL) level, performing a retrofitting of the software into a MIL environment.

Table 3 compiles the input cases performed including a brief description and the expected outcome. First, some nominal cases are defined, in different launchpads, to check that no termination condition is triggered, assuming that the flight is safe throughout the entire trajectory. The next step consisted of creating input cases to test all the mission rules and outputs implemented, using Andøya as reference launchpad.

Table 3. Test cases for Mission Rules validation

Scenario	Description	Outcome
Andøya nominal	The launcher reaches orbit describing a safe flight	EoR
Bowen nominal	The launcher reaches orbit describing a safe flight	EoR
Kourou nominal	The launcher reaches orbit describing a safe flight	EoR
Engine loss	The nga falls to zero to simulate the engine loss and detect tumbling	Terminate
Engine failure	Noticeable perturbations applied to nga to cause path deflections	Terminate
Tracking input loss	No valid tracking inputs reception is simulated	Terminate
Boundaries	The IIP estimation falls out of the allowed flight boundaries	Terminate
Hierarchy	Used to check that safe condition overrules termination	SC

The engine loss test is part of the validation campaign. In this case, the acceleration profile falls to zero due to an engine cuts-off. The tumbling scenario is detected because the launcher starts losing altitude, but the Instantaneous Impact Point (IIP), the touchdown point over the Earth surface, remains almost in the same position. The expected outcome is a termination signal. The described trajectory and the detection point are shown in Fig. 10.

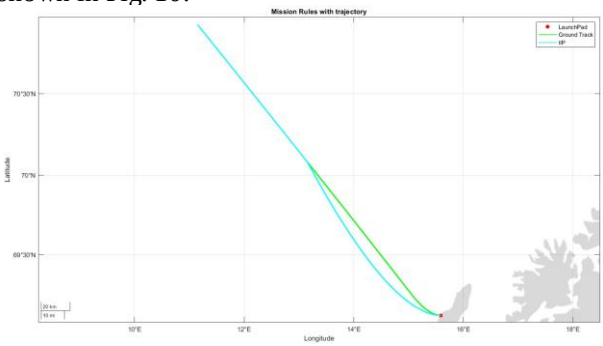

Fig. 10. Tumbling detection at engine loss scenario

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

Moreover, Fig. 11 represents the altitude variation against the downrange variable. It is differenced the instant when the engine is lost and when tumbling is detected, since the algorithms are able to compute the delay between both events.

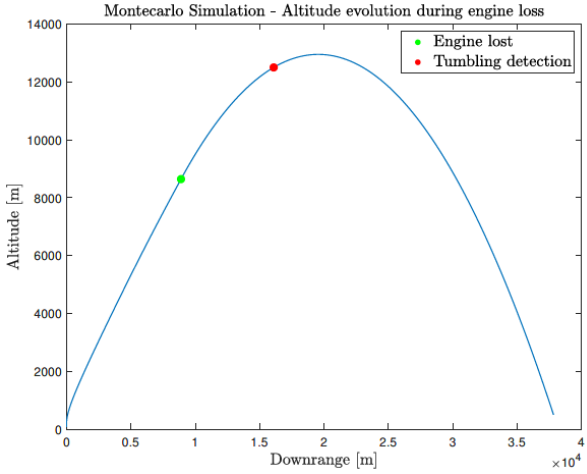

Fig. 11. Engine loss and tumbling detection

To visualized other types of results, the ones related to the boundary check are presented. The termination flag shall rise because the IIP (blue line) of Fig. 12 is out of the allowed flight region.

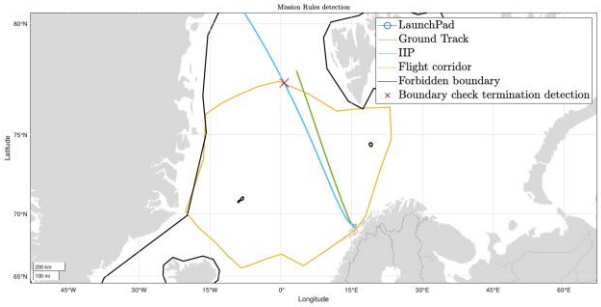

Fig. 12. Boundary check detection in Andøya scenario

Fig. 13 shows the behavior of the termination flag. The transition occurs when the launcher leaves the defined flight corridor (yellow region).

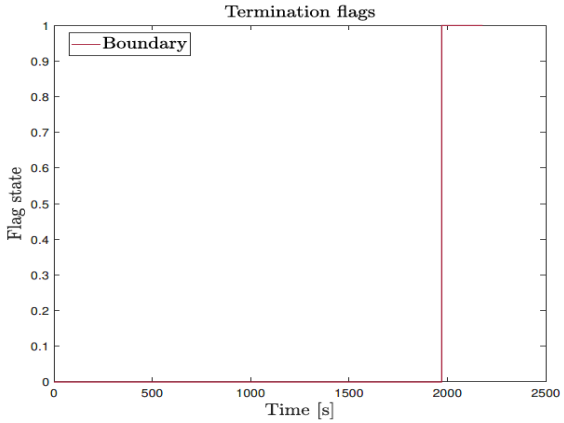

Fig. 13. Termination flag for boundaries scenario

To conclude the results section, it is included a Montecarlo simulation to represent the engine failure test case. The nominal trajectory is perturbed introducing deviations in the y-axis of the non-gravitational acceleration profile. It allows to check extreme cases, considering positive and negative factors.

Fig. 14 depicts the IIP projection of different trajectories follow by the launcher when the perturbations are applied. As outcomes, a SC signal is obtained when the launcher leaves the flight region through the corridor gate (horizontal blue line). In contrary case, the result achieved will be a terminate.

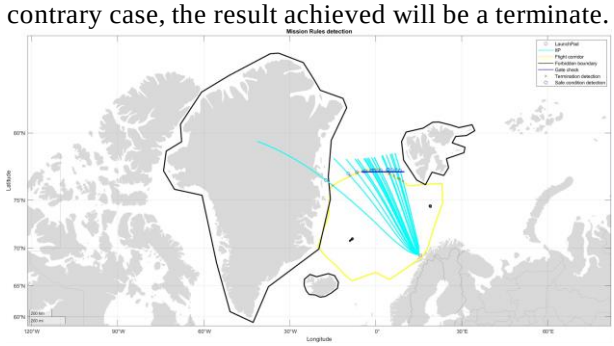

Fig. 14. Montecarlo results for engine failure scenario

As a future directive, the mission rules and navigation algorithms will be integrated into a flight platform to validate at hardware-in-the-loop (HIL) level and receiving data from an EGSE. This procedure will be useful to check the response time and the accuracy of the results.

4. Roadmap

SAFEST is an initiative to achieve the first operational version of an AFTU demonstrator with TRL 5. The experience acquired in previous projects has been leveraged for that purpose.

The roadmap has two main focuses of work: (i) proposing a maturation plan for missing technologies improving the capabilities with the corresponding validation; and (ii) the system-level testing and qualification including ground and flight campaigns.

The maiden flight is preceded by a set of models and tests. The prototype design ends up with an engineering model (EM) to perform the ground test campaign. In parallel to the EM progress, the ground functionalities (EGSE and MGSE) must be developed to check interfaces or environmental behavior, among others. The results will be used to evolve to the qualification model for flight test. Finally, the flight model is tested in a shadow flight, integrated in the vehicle but not as part of the safety chain.

The SAFEST project relies on SENER investment to support the AFTU development approach. Furthermore, some actions have been considered to promote it as

Copyright ©2024 by the International Astronautical Federation (IAF). All rights reserved.

identifying external funding sources and partnerships agreements, mainly with emerging launcher start-ups. The availability of a mature unit will facilitate the understanding with potential customers.

5. Conclusions

A more affordable access to space in terms of cost and flexibility is mandatory to meet user and market demands. A distinctive solution in this direction is the AFTS, which transfers safety processing operations on board the launcher vehicle.

This paper has presented the work performed so far for the development of a European AFTU device. The design of this unit is driven by the user needs identified before and during the project execution.

The main results obtained are:

- The proposed AFTU prototype and AFTS architecture is cheaper, more flexible and allows higher launch cadence than traditional FTS.
- Main features of SAFEST design, allowing to have a highly configurable and scalable design to different local regulations and launchers.
- All the mission rules algorithms work well for the different case scenarios presented. It is still pending the results with the fully integrated prototype.

Next steps of the development include the maturation and testing of several critical technologies identified, as well as the preparation of a functional EM model. With this objective, Sener has led an industrial consortium that have recently won an ESA project, called LoCRAFTS, devoted to further derisk AFTS technologies.

Acknowledgements

This work is carried out in the frame of the SAFEST Project, where Sener is the coordinator of a joint effort of many individuals and organizations. This project has received funding from the European Union's Horizon Europe research and innovation programmed under grant agreement No 101082662. We greatly thank of the great work of all the consortium members, the European Commission officers and reviews and the business development department from Sener, which have believed in and firmly committed to the project.

The preliminary design and architecture of AFTU was supported by several individuals who have left the company pursuing their dreams. In particular, we want to thank Rafael Polonio, Ugaitz Marcos and Javier Ruiz, for their valuable contribution to the project.

References

[1] RCC319-19 Flight Termination System Commonality. 2019

[2] Groves, Paul. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition. 2013. ISBN: 1608070050.

[3] S. Lozano, J. Fombellida, C. Rodríguez, C. Tato, J. Carretero, "MFOC Project: MPSoC-Based Multi-Purpose Execution Platform", III Congreso de Ingeniería Espacial: El espacio, la última frontera, Madrid, Spain, 2020, 27-29 October. ISBN: 978-84-09-31948-0. pp. 120-122.

[4] E. Geist, "Core Flight System Training - cFS Draco," Nasa.gov, Jan. 19, 2024. https://ntrs.nasa.gov/citations/20240000217 (accessed Sept. 26, 2024).

[5] M. Masmano, I. Ripoll, A. Crespo, J. Metge, "Xtratum: a hypervisor for safety critical embedded systems." In 11th Real-Time Linux Workshop, vol. 9, September 2009.

[6] "RTEMS Qualifies for the Space Domain | RTEMS Real Time Operating System (RTOS)," Rtems.org, 2022. https://www.rtems.org/node/139 (accessed Sept. 26, 2024)