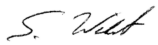
Next Generation Flight Termination System for Launchers

Executive Summary Report

Contract No.: 4000136253/21/NL/MG

Technical Officer: Stephan Schuster (TEC-MPA)

Document Number:	ASTOS-FTSnext-ES-001	**Date:**
Issue:	1.0	2023-07-25

	Name/Function	**Organization**	
Prepared by:	Julia Gente	Astos Solutions	
	Marc Hirth	Astos Solutions	
Checked by:			**Signature:**
Product Assurance:			-
Project Management:	Sven Weikert	Astos Solutions	

Document Change Record

Issue	Date	Affected Chapter/Section/Page	Reason for Change Brief Description of Change
1.0	2023-07-20	All	First issue

Table of Contents

1 Introduction

This document is prepared for the project “Next Generation Flight Termination System for Launchers” (FTSnext) lead by Astos Solutions under contract of the European Space Agency. It summarizes the objectives of the project and the work performed to achieve these objectives.

2 Applicable and Reference Documents

2.1 Applicable documents

[AD1] ESA-TRP-TECMPA-SOW-020174, Next Generation Flight Termination Systems for Launchers, ESA Statement of Work, iss. 1, rev. 3

2.2 Reference documents

[RD1] RCC 319, Flight termination system commonalty standard from the Range Commanders Council (US DoD, US FAA, US DoE, NASA)

[RD2] FAA, Advisory Circular 14 CFR 450.101 and 405.108, 2021

[RD3] FAA, 14 CFR PART 450 - Launch and Reentry Licence Requirements, Code of Federal Regulations, 2021

[RD4] Range Safety Group, Global Positioning and Inertial Measurements Range Safety Tracking Systems Commonality Standard 324-11, Range Commanders Council, 2011.

[RD5] “Flight safety code,” https://www.industry.gov.au/dataand-publications/flight-safety-code, 2019

[RD6] “Space industry regulations 2021, www.legislation.gov.uk/ukdsi/2021/9780348223682, 2021

[RD7] Centre National D'Etudes Spatiales, "Article 67 Objectifs du système de neutralisation," in Arrete portant reglementation de l'exploitation des installations du centre spatial guyanais, 2009, p. 57.

[RD8] J. Ahn and W.-R. Roh, "Analytic Rime Derivatives of Instantaneous Impact Point," vol. 37, no. 2, 2014.

[RD9] J. Ahn and W.-R. Roh, "Noniterative Instantaneous Impact Point Prediction Algorithm for Launch Operations," Journal of Guidance, Control and Dynamics, vol. 35, no. 2, 2012.

3 Terms, Definitions and Abbreviated Terms

3.1 Acronyms

The following abbreviations are used throughout this document.

Acronyms	
AD	Applicable Document
AFTS	Autonomous Flight Termination System
Astos	Astos Solutions
ESA	European Space Agency
FAA	(US) Federal Aviation Administration
IIP	Instantaneous Impact Point
IMU	Inertial Measurement Unit
INS	Inertial Navigation Sensor
GNSS	Global Navigation Satellite System
MBSE	Model Based System Engineering
RCC	Range Commander Council
RD	Reference Document

3.2 Terminology

The following terminology is used throughout this document.

AFTS

An Autonomous Flight Termination System (AFTS) is defined as an autonomous system composed of software and hardware that operates on-board of the space vehicle and is able to trigger and control the self-destruction of the launcher without human intervention. An AFTS includes all associated software, hardware, and subsystems, such as GPS receivers, GPS antennas, batteries, and INSs, used to make termination decision.

FTSnext

The Next Generation Flight Termination System (FTSnext) is defined as the concrete instantiation and development of an AFTS that is designed in the presented documents. It does not contain the explosive termination devices.

4 Project Objectives

The main objectives of this project were the design, simulation, and functional verification of an Autonomous Flight Termination System (AFTS) for launchers [AD1]. This AFTS shall allow the flexible use on any launch site and any launch system with minimal need for adaptation, while conforming to safety regulations and constraints. The primary application shall be the upcoming generation of European (micro)launchers, but also a retrofit to existing systems shall be considered as an option.

It was split into the following objectives:

- Performing of an analysis of the regulatory frameworks corresponding to launch safety and flight termination.
  - Commonalities, but also differences between the country's legal launch safety frameworks shall be assessed.
- Derive the requirements for future AFTS.
  - The current generation of (non-autonomous) flight termination systems (FTS) and flight safety systems (FSS) shall be analysed in detail.
  - A consistent and complete set of requirements shall be created.
- Realise the subsequent preliminary design of a next generation flight termination system for launchers.
  - Improving the functionality over the current existing flight systems, while maintaining the required safety standards.
  - Comparisons between the traditional and the proposed next generation FTSnext shall be made, and differences shall be quantified.
  - Necessary interfaces to launch vehicle, ground stations and orbital GNSS constellations shall be evaluated and required assets on the vehicle, on-ground or in-orbit shall be defined.
  - Aspects relevant for mission performance and vehicle integration (e.g., mass, power consumption, thermal aspects, interference) shall be addressed.
- Realise an implementation of the proposed algorithm, simulation of the environment, and functional verification of a next generation flight termination system for launchers.
  - System algorithms shall cover initialization, launcher localization as well as launcher state and mode monitoring.
  - The algorithms of the launcher state and mode monitoring shall be designed with the aim to detect and possibly predict upcoming problems, deviations or malfunctions in the flight of the vehicle.
  - An exhaustive list of parameters, states and modes necessary to be monitored during launch to be compliant with the safety regulations needs to be derived and implemented into the algorithms.
  - The software shall follow a 'plug-and-play' approach, hence a flexible application on different launch vehicles and launch sites should be feasible.
  - The simulation shall realistically mimic the vehicle launch and ascent (in an emulator environment) and the realistic consideration of all system functions and interfaces.
  - The software shall automatically identify non-nominal and critical behaviour and independently trigger the flight termination if it detects a rule violation.

5 Performed Tasks and Outcomes

5.1 Analysis of Regulatory Framework

Sixteen launch sites were contacted. Seven launch sites from the United Kingdom, Sweden, Japan, Germany, Portugal, France, and Australia provided information on regulations they apply. Most of them stated that they follow best practice guidelines from the United States Federal Aviation Administration (FAA). The remaining sites did not request special regulations.

The FAA regulations on flight abort can be found in the Title 14 Code of Federal Regulations (CFR) in part 450.108. The code gives general rules on flight abort systems. It addresses the necessary reliability, objective and constraints on flight limit, abort rules, and application requirements. The corresponding Advisory Circular [RD2] gives more detailed information on how to demonstrate compliance with the associated regulatory requirements from the CFR. It allows to develop flight safety limits and flight abort rules that can be implemented in an algorithm.

The Esrange Space Center in Sweden and SaxaVord space port in the United Kingdom requested to follow the Range Commander Council (RCC) Flight Termination Systems Commonality Standard 319 [RD1]. The standard gives an extensive description of design and testing requirements to validate FTS. It also considers special requirements on AFTS. RCC standard 319 for example requires that the FTS shall have a statistically predicted reliability with a 95% single-sided lower confidence boundary of at least 0.999. It requires to use redundancy of certain components to avoid single-point failures [RD4]. The same reliability is also required by CFR §459.108 b for flights with a number between 10-2 and 10-3 conditional expected casualties for uncontrolled areas.

Additional regulations were analysed including the United Kingdom (UK) Space Industry Regulations [RD6] and the Australian flight safety code [RD5]. The UK regulation does not cover specifications on FTS or AFTS. However, requirements are placed on the flight termination personnel. These can be transferred on the algorithm in an autonomous system. Furthermore, it generally asks to outline the use of FTS in the safety case and safety operation manual, which is necessary to obtain a launch license in the UK. The Australian flight safety code defines system performance specifications that correlate to RCC standard 319-14. However, in contrast to RCC, it does not specifically state that the system needs to be redundant.

French regulations generally require the possibility of manual termination by a range safety officer in addition to an AFTS [RD7].

5.2 Functionalities, Design and Requirements

FTSnext was developed using Model Based System Engineering (MBSE). The system was designed to fulfil the needs of the users as well as the legal regulations on flight termination.

MBSE was implemented in accordance with the Architecture Analysis & Design Integrated Approach (ARCADIA). It was used in the project to analyse the needs of users, authorities, and the system itself and to find a system architecture that meets these needs. The method allows to systematically define and validate software and hardware architectures. The development framework ARCADIA splits the system engineering into the four phases “Operational Need Analysis”, “System Need Analysis”, “Logical Architecture Design”, and “Physical Architecture Design”. The first two phases provide system needs in terms of

functions and capabilities that are derived from the user requirements. These phases are still open to all solutions and capture mainly the necessary functions that the system shall provide. Based on this function set a set of requirements is derived. In the last two phases a system architecture is built that provides the solution to meet all requirements and user needs in all scenarios of usage. The system design is structured in components and functions that are linked to them. The architecture covers hardware as well as software together with all system interfaces. The method allows traceability between the four levels of development. The method was implemented with the Eclipse Capella tool. The output of each phase was a set of block diagrams which also serve for documentation.

One main identified operational need of the AFTS is the termination of off-nominal flights whose continuation would impose an unacceptable risk to humans or assets. Secondly, adaptability to different launchers, launch sites, regulations and trajectories is desired. This makes it necessary to provide easy requalification as a capability of the FTSnext system.

In the functional analysis a set of more than 50 software, hardware, and user functions was obtained together with the functional exchanges. To operate an AFTS, two main software components are used. These are the onboard flight software and an offline software to configure and validate the software that is loaded to the onboard system. Main functions of the onboard system are to monitor the launcher flight and to evaluate termination or set to safe decision criteria. Safe mode means that termination is made impossible.

A subset of the identified functions on the launch operator side are necessary to provide data to customize the AFTS. This customization data covers launcher specific set up and nominal trajectory data. The safeguard authorities need to provide information on the necessary termination rules and the requirements that are used to validate the system. The ground station network is less important than in the traditional FTS, but it is still used for the telemetry of monitored data to the ground. The functionality to receive termination commands from ground control is optional.

Over 240 requirements were derived from the regulations, state of art, and MBSE. All requirements were associated to a corresponding system component. This allowed traceability from the system design to requirements and vice versa.

The physical design was developed in accordance with the system design and the requirements. It consists of the following parts:

- Inertial Measurement Unit (IMU)
- Global Navigation Satellite System (GNSS) receiver and antenna
- Processing units
- Safe and Arm device
- Termination unit
- Power unit
- Telemetry unit

Redundancy is used to provide the necessary reliability. Processing unit and power units are redundant. Furthermore, two navigation systems provide position and velocity estimates. These are the GNSS and an inertial navigation system that estimates position and velocity from the IMU measurements. During full sensor performance, and if the IMU and GNSS data do match, the measurement data is fused. Sensors were chosen to provide the accuracy of 0.3 m/s for velocity and 100 m for position (1 sigma interval) required by RCC 324 [RD4]. At the end of the project, it is not possible to provide information on the reliability of the system. Due to the modular approach reliability can be increased by changing hardware or increasing redundancy, if needed.

5.3 Algorithm Design

The FTSnext software is split into onboard and offline software.

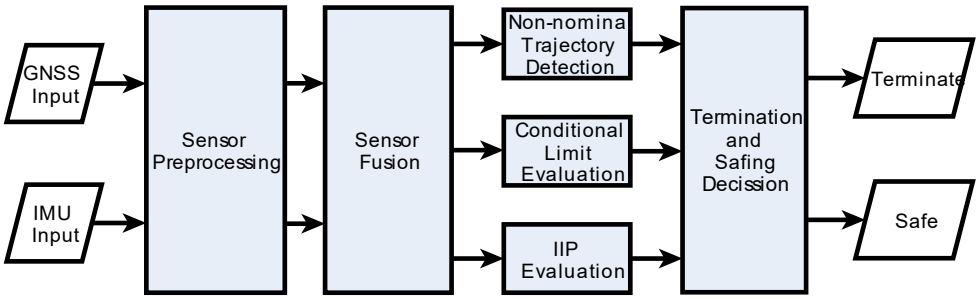


Figure 5-1: Excerpt of onboard FTSnext software architecture. Grey rectangles denote software subfunction. White rhomboids represent the inputs and outputs of the software. Arrows represent exchanges.

The onboard software provides flight monitoring and termination or safe mode decision. A block diagram of the onboard software architecture is given in Figure 5-1. Sensor preprocessing allows the software to be robust against unhealthy sensor measurements. That allows the system to take advantage of the redundancies and cover outages. Sensor fusion improves the measured variables to one solution that best represents all measurements. It also performs sensor cross validation.

The determination of flight criticality is split into three subfunctions that assess different critical flight situations. In the non-nominal trajectory detection states are directly evaluated against general limits that allow for a useful mission. The Instantaneous Impact Point (IIP) evaluation shows whether the vehicle is contained inside a controlled area. Another tool to detect unhealthy trajectories are so called conditional limits. They are introduced by the advisory circular of FAA [RD2]. In contrast to other flight limits, the conditional limits are evaluated only at certain time points during the flight or after singular events. The results of all evaluations are passed to the termination and safe mode decision unit. The termination rules that were chosen as an orientation for the project are:

- Unacceptable risk to protected area
  - IIP latitude and longitude, as compared to a pre-established polygon
  - Straight up launch: Max altitude vs downrange distance
  - Impact with propellent: Minimum altitude vs downrange distance
- Vehicle behaviour is outside validated bounds
  - Acceleration
  - Angular rate
  - Data loss duration
  - Sensor quality
- Impossible to reach orbit
  - Conditional limit (e.g. perigee height or flight time when leaving monitored IIP corridor)
- Unsafe to overfly unprotected area

- Conditional limit (e.g. IIP velocity or time-to-impact while IIP traverses area)

The system design is generally independent of the termination rules, as these may vary depending on the decisions of local regulators. However, it is necessary to have an idea on what can be required so it is possible to decide on the set of states that will be evaluated onboard. The following states were chosen to monitor:

- Position
- Velocity
- Angular rate
- Acceleration
- Instantaneous Impact Point
- Sensor availability and quality

The offline software is used to customize, simulate, and qualify the onboard software. Customization of the software is provided by parametrization and selectable wrapper functions. Wrapper functions can be reimplemented to make the system compatible to hardware or regulations. In contrast, core functions are present in every instance of software. After customization, simulations and tests must be conducted to show that implemented rules provide all necessary functionalities.

5.4 Simulator and Software Implementation

To show the feasibility of the system design, the FTSnext software was implemented within the MATLAB environment and programming language. Simulink was used to build the Simulator that contains the prototype of the software.

Several flight trajectories were created in ASTOS. The launcher dynamics are simulated in it, and corresponding sensor measurements generated. Three nominal scenarios are created of representative launchers. These test cases differ in launcher dimensions, trajectories, and launch sites. The created nominal trajectories are used in conjunction with simulation of failure trajectories to validate the function of the FTSnext software and to show that the termination is triggered whenever required. Simulated failure cases are:

- Turn towards impact line
- Tumbling
- Straight up launch
- Extreme acceleration
- Not reaching orbit.

The onboard software prototype was implemented in line with the software architecture.

Interfaces to GNSS receiver, IMU sensor, umbilical, and termination unit are simulated and part of the prototype.

The FTSnext monitors the sensor quality. It ensures that the evaluation of the flight trajectory is based on healthy measurements. In a first step, basic health checks are performed in the sensor preprocessing module. This comprises determination of the sensor communication status, detection of stale or frozen sensor data as well as measurement consistency checks (e.g., versus expected data ranges and jumps). Resulting measurement quality and validity flags are fed as input to the sensor fusion module for further processing.

An extended Kalman filter was implemented as a sensor fusion. Angular rate and acceleration IMU measurements are propagated in an inertial navigation that is further fused with GNSS position and velocity measurements. Sensor biases are estimated by calibration at the launch pad. The navigation algorithm can bridge temporary sensor outages of either GNSS or IMU by relying on healthy measurements. The residual of the update step is evaluated to detect if IMU and GNSS measurements are consistent with each other. If the residual exceeds a limit the status of the estimate is set to invalid. In this case inertial navigation solution and GNSS measurements are both independently passed to further criticality determination of the trajectory.

The sensor fusion features an extensive error handling. It considers the availability of GNSS position, GNSS velocity, IMU angular rate, IMU acceleration, outage duration, and flight phase. Depending on the currently available sensors, the outage is handled by relying on the remaining measurements. The sensor fusion quality is estimated and forwarded to be considered in the termination decision. The estimated trajectory states are inputs to the modules monitoring the trajectory health (with respect to the defined flight safety limits), namely non-nominal trajectory detection, conditional limit evaluation, and IIP evaluation.

The latter predicts the IIP based on the current estimated position and velocity state of the launcher. The implementation is based on a Keplerian algorithm which accounts for the oblateness of the Earth [RD8] [RD9]. The algorithm neglects drag-effects, but it is well-suited for onboard-implementation due to its non-iterative nature. The IIP evaluation then further determines the location of the predicted IIPs with respect to user-defined polygon areas that either represent no-go areas for the trajectory or specific areas that trigger additional checks of safety limits (e.g., when overflying populated islands or land).

The conditional limit evaluation and non-nominal trajectory detection modules are used to detect abnormal launcher behaviour during the ascent. In the latter case, user-defined profiles (e.g., altitude over downrange distance, angular velocity over time) are continuously monitored and compared with respect to predefined limits (minima, maxima or bounds). The conditional limit evaluation operates in a similar manner by applying limit checks, but it is only executed when specific conditions along the trajectory (‘gates’) are reached or when the IIP enters a conditional limit area as described above.

In both modules, the gate or profile abscissa and ordinate data can be chosen generically from all relevant onboard data i.e., time or flight-time, estimated states (e.g., position or acceleration) or derived states (e.g., downrange distance, perigee height) to allow maximum flexibility. The number of defined gates or profiles is customizable as well.

The information from the previous modules is finally collected and evaluated in the termination decision module.

In general, a termination is triggered if:

- The predicted IIP leaves the predefined area of permitted IIP ranges.
- A limit check for any of the predefined profiles fails.
- A limit check for any of the predefined conditional limits fails.

To avoid termination due to a single-point event, the consecutive and overall duration of the termination conditions as well as the maximum overall occurrence can be configured, and respective timers are implemented to delay the ultimate determination.

As all termination conditions fully rely on the estimated launcher states, also the measurement data availability and quality affect the termination decision – such that after predefined periods of entire data loss or insufficient quality of the sensor fusion estimates, a termination is triggered as well. On the other hand, safe mode is triggered to avoid

termination under undesired conditions. This applies for instance on the launch pad or once an orbit is reached.

5.5 Test Campaign

Several test cases were implemented to show the compliance of the prototype with the requirements.

The test campaign includes the three baseline scenarios of two micro launchers and one Ariane 5 launcher. They verify that - though artificially designed - all conditions to be monitored by the AFTS are set up and processed correctly. They also show the adaptability of the algorithm to various launch sites trajectories. Due to a lack of a full envelope of possible states and IIP locations, which result e.g., from a 'Trajectory Analysis for Normal Flight' according to FAA regulations, permitted IIP bounds, conditional limits, and bounds for dynamic launcher states were artificially constructed. They are violated for the specific failure cases while being met for the nominal and near failure scenarios.

Seven failure scenario tests are designed to show that the AFTS detects and reacts appropriately to the identified failure cases. They represent cases of slow and fast trajectory turn towards the permitted impact line, tumbling, straight-up launch, extreme acceleration and no lift off. Apart from the no lift off case all scenarios shall trigger termination. For these cases the results show that the AFTS correctly detects the termination rule violations and triggers a termination after the respective timer is reached. In case of no lift-off, it is shown that the AFTS neither triggers a termination, nor a safe-lock is set in the AFTS.

Four near-failure scenario tests are designed to show the AFTS reaction when the trajectory closely approaches the permitted boundary without leaving it. They are a small trajectory turn from the nominal trajectory, higher acceleration, orbit just reached and low thrust. It is shown that the AFTS does not show any termination indications as all conditions remain met – though marginally. At the end of the simulation, a permanent safe-lock of the AFTS is set.

The performance of the sensor fusion was assessed with four test cases. Monte Carlo simulations were conducted to test the navigation and its sensor outage handling. It was used to calculate the accuracy of the position and velocity estimate. The simulation used the trajectory of a nominal micro launcher scenario in 500 runs. The sensor error model, initial state estimate, estimated gravity, and Earth model were varied from run to run. The obtained position and velocity accuracy meet the requirements of 100 m and 0.3 m/s in the 1 σ interval. Another 100 Monte Carlo runs were conducted to test the sensor outage handling of the navigation algorithm of outages that do not require termination. The simulation triggered random sensor outages of position, velocity, acceleration, and angular rate. The increase in position and velocity estimate error are still within the acceptable bounds. The navigation solution fully recovers from the outages. Two further test cases were implemented that show the sensor fusion can handle high vibrations and acceleration. In the vibration test white noise is applied on the acceleration measurements with spectral density of 0.04 g^2/Hz. For the acceleration case, a period of 5 minutes with twice the maximum nominal acceleration value is simulated. In both scenarios the state estimate accuracy meets the requirement.

Another set of simulations was dedicated to test the accuracy of the IIP prediction. For that purpose, results of the implemented analytical algorithm were compared to the impact points resulting from ‘free-fall’ trajectories with varying initial states taken from the nominal launcher states during the ascent. These trajectories were generated with a 3 degrees of freedom simulator using different gravity model orders and with a simplified drag model

(exponential atmosphere and constant ballistic coefficient) enabled/disabled. The vacuum impact accuracy requirement in RCC Standard 324 which is constant up to 20 km impact distance and increases linearly with larger distances can safely be met for all vacuum impact cases. Violations due to neglected air drag range up to about 10 % of the impact range for impact distances between approximately 1-100 km. However, this does not pose an issue if the same (vacuum impact) model is used to lay out the permitted IIP bounds for a launch scenario according to FAA guidelines.

The remaining test scenarios analyse the compliance of specific functional units with the requirements:

- Data loss handling scenarios showing that termination is triggered in cases of unacceptable sensor outages (unless the outage occurs at the launch pad):
  - A short outage of acceleration and GNSS measurement
  - A long GNSS outage
  - A long angular rate outage
  - A long acceleration outage
  - Outage at the launch pad
- Cross validation scenarios showing that if cross validation fails, the FTS switches to the raw GNSS solution and the INS estimate with the following expected reaction:
  - No termination is triggered if none of the two solutions indicate termination. Even if only one navigation solution is available (maximum time period can be defined)
  - Termination is triggered if either of the solutions violates a termination rule
- Scenarios showing that a termination can be avoided in case of a single-instant termination condition. The termination timer is used with configurable limits and counters.

Furthermore, a set of unit tests was created that provides full code coverage of the FTSnext algorithms. The unit tests focus on the main functionalities of the written functions.

6 Conclusion

The FTSnext project provides a broad overview of regulation on autonomous flight termination systems. A vast set of requirements has been generated from the regulations. An adaptable system design that fits these and is flexible to suit varying international and flight specific circumstances has been developed. Clearer national or preferably international regulations are desirable to minimize the need for requalification between launches.

The implemented simulator and FTSnext software prototype are a valuable starting point in the development of international AFTS. First steps in the validation of the system design have been taken and all applicable requirements are validated with the implemented prototype. In the next steps the prototype should be used to perform a reliability analysis of the system to evaluate that it can provide statistically predicted reliability with a 95 % single-sided lower confidence boundary of at least 99.9%. Based on that, the hardware prototype can be developed and tested.