DOI: 10.13009/EUCASS2022-4841
9TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Missionisable on-board flight safety based on real-time autonomous decision-making for microlauncher services

Nil Martín[1], Carla Navarro[1], Eduard Díez[1], Alejandro Sabán[1], Jordi Martín[1] and Magda Escorsa[1]
[1] GTD SSII, Passeig Garcia Faria 17 (08005 Barcelona, Spain)
nil.martin@gtd.eu; carla.navarro@gtd.eu; eduard.diez@gtd.eu; alejandro.saban@gtd.eu;
jordi.martin@gtd.eu; magda.escorsa@gtd.eu

Abstract

The change of the space industry towards NewSpace and its consequent needs for new adaptative launch systems sustains the development of innovative FSW solutions. This paper presents an autonomous on-board safety based on fuzzy logic aiming to address multiple launcher configurations and mission profiles to reduce mission associated costs. A Flight Manager state machine is introduced to configure the FSW optimising it for the requirements of each flight phase, using semi-fuzzy state transitions aiming to adapt mission timeline against small perturbations. The proposed safety system is compatible with the multiple paths of development for future spaceports. The combination of a modular architecture and the flight manager provides with a high configuration capacity to particularise FSW for multiple missions, contributing to the reduction of mission preparation times.

1. Introduction

Over the last 20 years, miniaturization and New Space have appeared as the two leading trends in the field of satellite technologies and have therefore driven an important reshaping of the space launch market. Launches recently made by OneWeb and SpaceX (Starlink constellation) make patent the change that the space industry is facing. Indeed, these small satellites can now provide many orbital services, such as Internet, Earth observation and scientific research, among others, which were historically provided by larger, heavier satellites. As a matter of fact, small satellites have been the payload of one third of the launches made in the last two decades [1] and present an increasingly growing pattern, with studies forecasting commercial operations to be the 80% of the space access demand [2].
These SmallSats can be launched as the piggyback payload on rockets used for large satellite launches, giving them no decision on the orbital and launch parameters, on rides to the International Space Station (ISS) through resupply missions and placed into orbit via its CubeSat deployer, or in ride-sharing operations of large sets of micro and nano-satellites servicing different customers, an option that can be difficult to coordinate.
It can be easily envisaged that these launch alternatives are far from adequate, hence New Space satellites see their feasibility hindered by the non-optimality of the existing launch opportunities and claim for innovative solutions. Continuing with New Space challenges, a general reduction in both launch and production costs is needed for new and more actors to enter the sector. However, this cost reduction cannot be inimical to reliability. This has originated the opportunity for the development of newly adapted launch services targeting a set of needs to be fulfilled: orbital flexibility, multi-payload capacity, higher launch frequency, shorter launch campaigns and a reduction in costs without compromising reliability and performance. These leads to the development of the so-called microlaunchers.
The innovation in the domain of the Flight Software (FSW) enormously contributes to above scenario in terms of flexibility and missionisation. Specifically regarding FSW devoted to guaranteeing the safety of the mission, on-board autonomous flight safety systems (AFTS) have been under study and development for the last decade on most New Space initiatives, and have even reached the heavy launcher services, with the objective of reducing ground infrastructures and interfaces, operations campaign duration and inter-campaign preparation, such as training dedicated to the safety mission, having a direct impact on costs. In terms of performance, on-board safety systems reduce the decision-making chain and increase the flexibility of the launch service through on-board FSW missionisation. Therefore, the development of innovative technologies for the AFTS is crucial to meet the New Space necessities and exploit the micro-launchers market.

DOI: 10.13009/EUCASS2022-4841

This study is integrated in the frame of activities of the H2020 project ENVOL (funded by the EC Grant Agreement 870385 lead by NAMMO), a project devoted to the design and demonstration of a hybrid propelled micro-launcher targeting LEO and SSO orbits at 800km providing services to payloads up to 200 kg, in which GTD is responsible for the design and prototyping of the avionics and Flight SW.

2. Proposed Approach

Within the introduced scenario, this paper proposes a modular and configurable On-board Autonomous Safety SYStem (OASSYS) constituted by a diagnosis unit enabling to monitor mission and vehicle health status and a decision-making unit that substitutes the ground range officer in charge of issuing the flight termination signal. The decision-making component requires awareness of mission and vehicle conditions as an input to apply the configuration that best fits the specific requirements of the flight for every instant. Thus, a Flight Manager providing with these inputs with respect to the mission is proposed, not only for the OASSYS but as responsible for managing the whole FSW and adapting it to the optimal configuration for each mission phase.
The development of these modules is conducted in the frame of a modular FSW validation tool formed by an OBC running the FSW itself and a Software Validation Facility (SVF) capable of emulating a wide range of flight conditions, launcher configurations and launcher subsystems [3].

2.1 Safety

As mentioned, the proposed safety system is composed of two sequential phases, the diagnosis, where the scenario is assessed, and the evaluation, where the resolution decision is made.

2.1.1 Diagnosis modules

This set of modules are in charge of evaluating the dynamic state of the vehicle, as well as its internal health status, to compute the status of the vehicle itself, with the Integrated Vehicle Health Management (IVHM), as well as the status of the mission in terms of danger to third persons and goods (Impact Area and Flight Corridor). However, the article focuses on the repercussion of the impact area to mission safety and the developments of the IVHM and the flight corridor are out of scope.

Impact Area

This module computes the area of impact on the ground of the launcher in case of critical failure producing a non-nominal re-entry of the vehicle and its associated level of danger.
The impact point on ground is computed at each timestep with the vehicle's dynamic state as starting point. The state of the vehicle is computed by means of hybrid navigation (hybridizing IMU and GNSS sensor measures) [4] from a segregated chain of sensors dedicated exclusively to safety purposes. The dynamic state of the vehicle and its associated covariance matrix are then propagated on a ballistic fall to the ground using the F&G algorithm [5]. The propagated covariance at the impact point is then used to generate a two-dimensional ellipse representing the error in latitude and longitude of the impact point and hence, delimiting the impact area.
The assessment of the impact area is based on polygon clipping algorithms that allow to compute the intersection of two given polygons, the impact area and a mission defined protection area. Consequently, the driver used to assess the level of danger (LOD) is the ratio between the impact area in danger and the whole impact area [see Eq. (1)].

$$\boldsymbol{LOD = \frac{A_{danger}}{A_{impact}}} \tag{1}$$

However, as depicted in Figure 1, two possibilities exist for the definition of the protection area, which vary depending on the characteristics of the mission. For vertical launches, the spaceport operators define an Impact Limit Line (ILL) around the launch pad, creating a closed safety polygon or safety corridor inside of which an impact of the launch vehicle is not hazardous for people and goods. For these situations, the danger situation arises when the computed area is partially or totally outside of the ILL [see Figure 1a]. Since the clipping algorithm returns the intersection polygon between the impact area and the ILL, the intersected area corresponds to the portion of impact area in safety conditions ($A_{intersectio} = A_{safe}$). Furthermore, since $A_{impact} = A_{safe} + A_{danger}$, Eq. (1) becomes

DOI: 10.13009/EUCASS2022-4841

$$\boldsymbol{LOD = 1 - \frac{A_{intersection}}{A_{impact}}} \quad \textbf{(2)}$$

On the other hand, other type of missions such as those with an airborne launch define a set of protected zones where a potential impact might cause damage to people and goods. In these situations, as opposed to the prior case, the danger is found when the impact polygon is partially or totally inside the defined safety polygons [see Figure 1b]. Therefore, the intersected polygon returned by the clipping algorithm is directly the portion of impact area in danger ($A_{intersection} = A_{danger}$) and Eq. (1) simply becomes

$$\boldsymbol{LOD = \frac{A_{intersection}}{A_{impact}}} \quad \textbf{(3)}$$

In both cases, the returned value is within the range [0 1]. Nonetheless, since the ENVOL project is a vertical launch, testing procedures in the scope of this article will focus on the first case with a safety corridor defined around the spaceport.

(a) *(b)*

Figure 1. Representation of the potential situations to be encountered regarding the impact area. Configurations with a defined safety corridor (a) or a set of protected zones (b). Darker areas illustrate the zones where vehicle fall is forbidden.

2.1.2 Decision-making module

Decision-making algorithms appeared in the 1970s and have been gaining popularity and importance ever since. Nowadays, these algorithms are present among many disciplines in the forms of machine learning and artificial intelligence and a wide variety of techniques are available.
According to the operational requirements of the module, fuzzy logic appears, among the studied alternatives, as the best fitted type of decision-making algorithm [see Table 1]. The use of a fuzzy logic approach enhances the flexibility and missionization of the system allowing it to adapt to specific mission requirements.

Table 1. Decision analysis and resolution table for the evaluation of decision-making algorithms to be used at the core of the FLOS module.

Criteria	Driver	Weight	Neural Networks	Bayesian Networks	Expert Systems	Fuzzy Logic
Simplicity	RAMS improvement	20%	2	7	5	7
Configurability	Missionisation	35%	5	6	1	9
Configuration rapidity	Reduction of campaign duration	15%	2	6	1	7
Reliability	Autonomous System	20%	7	7	9	7
Cheap computational cost	Embedded System	10%	4	6	8	7
Score			**4.25**	**6.40**	**4.10**	**7.70**

Due to its criticality in a mission, the FLOS is designed in a modular architecture allowing for simplicity as well as easiness for a future scalability of its capacities. As graphically represented in Figure 2, the internal execution process is composed of three phases, the fuzzification of the diagnostic inputs, the evaluation of the status and the defuzzification of the obtained fuzzy set to get the crisp terminate or not-terminate outcome.

DOI: 10.13009/EUCASS2022-4841

Furthermore, the module incorporates a configuration block that allows the tuning of the fuzzy inference system parameters to adapt to the optimal configuration required by the mission phase being flown at each moment.

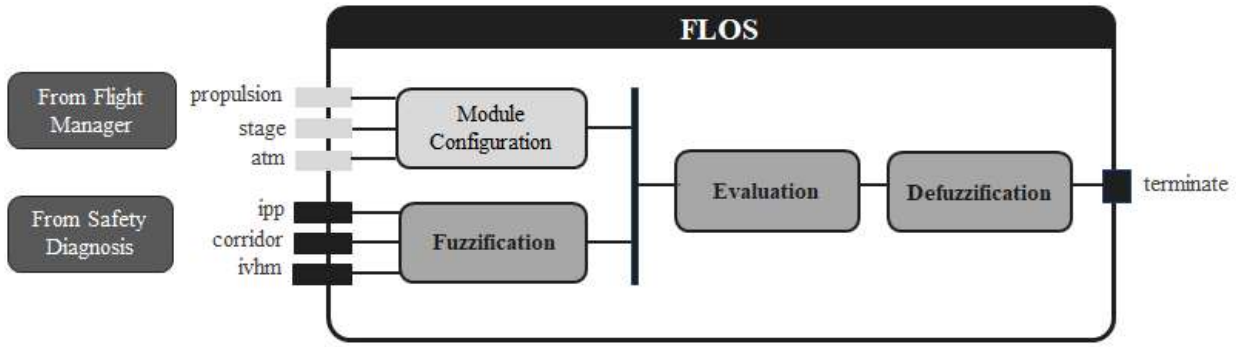


Figure 2. Block diagram of the internal FLOS structure.

In fuzzy logic, contrary to the classical set theory, elements do not follow the principal of bivalence[1] but are associated to a gradual and continuous membership in the unit interval [0 1] that assesses the degree of enrolment of the element to a certain quality or tag. These functions defining the quality/tag are called membership functions and play a fundamental role in the behaviour of the system, being decisive to obtain the desired performance of the module.
For this application, membership functions used for the fuzzification of each input are described in Table 2 and represent the level of danger that each diagnosis module has encountered in its evaluation. For each input, a single membership is implemented through a spline-based S-shaped function [see Figure 3]. These functions are characterized by two parameters (a, b) that correspond to the x-axis values at which limits of null (0) and maximum (1) membership is reached. These parameters correspond to the definition of null and maximum danger defined for each diagnosis block.

Table 2. Description of the membership functions used for the fuzzification of the diagnosis modules outputs.

Diagnosis module	Tag	Description
Impact Area	danger	Percentage of the predicted impact area in intersection with protected areas.
Flight Corridor	danger	To be defined.
IVHM	danger	To be defined.

Outputs membership functions are designed through an iterative trial and error process until successful decisions were achieved. A secondary membership function addressed to the inactive terminate is necessary to avoid indeterminate situations causing false decisions.

Table 3. Description of the membership functions used for the fuzzification of the FLOS outputs.

Output	Tag	Description
Terminate	active	Conditions to trigger an FTS command are met
	inactive	Conditions to trigger an FTS command are not met

[1] The principle of bivalence articulates that every proposition can only be either true or false, which are mutually exclusive. Therefore, a proposition cannot be true and false as well as it cannot be either true nor false.

DOI: 10.13009/EUCASS2022-4841

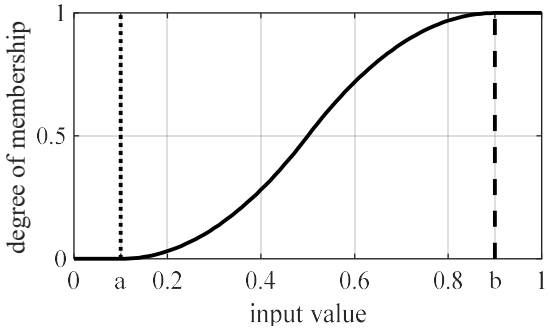


Figure 3. Example of an S-shape function defined by parameters a=0,1 and b=0,9. Axes are dimensionless.

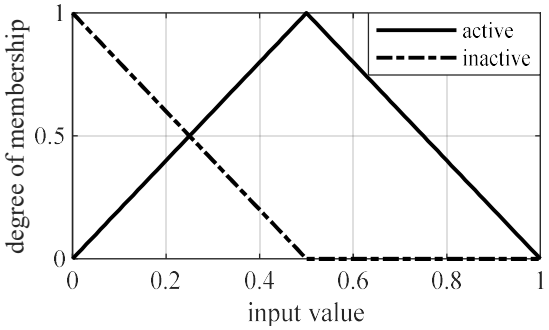


Figure 4. Terminate output triangular membership functions active [0, 0.5, 1] and inactive [0, 0, 0.5]. Axes are dimensionless.

The evaluation phase is governed by a set of rules defining the behaviour of the system. In formal or mathematical logic, a logic rule or rule of inference is a statement that take *n* premises and returns a conclusion, so that if the premises are true, so is the conclusion. However, fuzzy logic allows the premises and conclusions to be partially true, in concordance with the essence of the fuzzy theory.

Consequently, the set of rules implemented for the FLOS disclose the relations between inputs and outputs and hence, the behaviour of the fuzzy inference system. For sake of simplicity, rules combining multiple input or output memberships are discarded and only a single rule is set for each input-output combination, leading to the rules described in Table 4. Each rule has an associated weight in the range [0 1] which indicates its prevalence among the considered set. These weights are dependent on the mission requirements, allowing safety missionisation by means of its tuning.

Table 4. Definition and weighting of the rules governing the behaviour of the fuzzy system.

Rule	Weight
If Impact Area is dangerous then terminate is active	[0 1]
If Impact Area is not dangerous then terminate is inactive	[0 1]
If IVHM is dangerous then terminate is active	[0 1]
If IVHM is not dangerous then terminate is inactive	[0 1]
If Flight Corridor is dangerous then terminate is active	[0 1]
If Flight Corridor is not dangerous then terminate is inactive	[0 1]

Lastly, defuzzification is the process of converting the aggregated fuzzy set obtained at the evaluation phase into a crisp value for decision-making. The centroid method, which obtains the crisp number by computing the centre of gravity along the x-axis of the fuzzy set geometric figure [see Eq. (4)] is the most widely used defuzzification method and is the one chosen for the current application. The resulting crisp value (*cv*) is transformed to a binary signal by the direct application of the expression $cv \geq 0.5$, and triggers the FTS activation command if true.

$$\mathbf{x_{centroid}} = \frac{\sum_i \boldsymbol{\mu}(\boldsymbol{x_i})\boldsymbol{x_i}}{\sum_i \boldsymbol{\mu}(\boldsymbol{x_i})} \tag{4}$$

2.2 Flight Manager

The Flight Manager (FM) module is born due to the need of configuring the FSW, specifically safety, to adapt it to the most optimal configuration as function of the flight phase being flown. To achieve this management purpose, a finite-state machine is proposed and explained hereafter, capable of simultaneously managing the fields of interest for GNC and Safety applications. Nonetheless, the design of a finite-state machine allows to further exploit the FM and enhance its functionalities by adding a degree of intelligence on its interstate transitions to transform mission management into a more flexible concept capable of fighting against moderate external perturbations.

2.2.1 State-machine

A state-machine with high-level parallel states that allows to simultaneously and independently assess the multiple fields of interest (operative stage, propulsion system, atmospheric conditions, safety status and orbital operations) while guaranteeing design and validation simplicity.

DOI: 10.13009/EUCASS2022-4841

1. **Stage**. Formed by four sub-states (*idle*, *stage 1*, *stage 2*, *orbital module*) is in charge of indicating to the FSW which stage of the vehicle is operative at each moment of the flight.
2. **Propulsion**. Formed by three sub-states (*idle*, *on*, *off*) indicates the status of the propulsion system. The default entry state *idle* transitions to *off* when *Stage.idle* is exited. The *on* and *off* states are transitioned with the ignition and cutoff events as programmed in the mission timeline while guidance operates in open loop and obeying the orbital manoeuvres while guidance is running in closed-loop.
3. **Atmosphere**. Formed by two sub-states (*endo*, *exo*) specifies whether the vehicle is flying in presence or absence of atmosphere. Its bidirectional transition is ruled by the Kármán Line altitude (100 km), being *exo* above it and *endo* below it. The entry of the *exo* state triggers the release of the fairing.
4. **Safety**. Formed by two sub-states (*safe*, *terminate*) indicates the state of the mission as dictated by the safety module and is in charge of stopping the state-machine if entering the *terminate* state, which transition is unidirectional and it is accessed only when a termination command is issued from safety.
5. **Orbital Operations**. Formed by three sub-states (*open-loop*, *closed-loop*, *mission-end*) governs the guidance module to manage the orbital operations to be made and sets the deploy signal to the payload when injection conditions are met. After the execution of the last programmed manoeuvre, it transitions to *mission-end* which sends the signal to stop the FSW.

To activate the transitions between its internal states, the machine is fed up with the set of inputs described in Table 5.

Table 5. Description of the inputs required by the flight manager finite state machine.

Variable	Origin block	Description
Time	Clock	Time of the simulation
Mass flow	Launcher	Mass flow entering each engine of the whole vehicle
Chamber pressure	Launcher	Pressure on the combustion chamber of each engine of the whole vehicle
Navigation state	Navigation	Current kinematic state computed at the Navigation module
Safety Decision	Safety	Current safety status (safe or termination required)
Manoeuvre end	Guidance	Current state of the guidance algorithm being executed
Next Manoeuvre	Guidance	Indicator of the next manoeuvre to be executed by the guidance algorithm
Eject on manoeuvre	Guidance	Boolean indicating whether a payload shall be ejected at the end of the manoeuvre being executed or not

The state-machine outputs are divided in two groups as function of its purpose and target block. The first group of outputs (denoted flags [see Table 6]) is sent to the On-Board Computer (OBC) for FSW configuration while the second group (denoted commands [see Table 7]) is forwarded to the SVF to actuate upon the vehicle and manager the flight timeline.

Table 6. Description of the flags outputted and targeted to the OBC for FSW configuration.

Flag	Type	Description
stage	*Enumeration*	Flag of the enumeration class Stage specifying the stage being operated at the current flight time.
propulsion	*Enumeration*	Flag of the enumeration class Propulsion specifying whether propulsion is active or inactive at the current flight time.
atmosphere	*Enumeration*	Flag of the enumeration class Atmosphere specifying whether the flight is in endo- or exo-atmospheric conditions.
navigation	*Enumeration*	Flag of the enumeration class Navigation specifying the optimal navigation filter to be used for the current flight phase.
manoeuvre_count	*Integer*	Counter of executed manoeuvres.

DOI: 10.13009/EUCASS2022-4841

Table 7. Description of the commands outputted and targeted to the SVF for vehicle update.

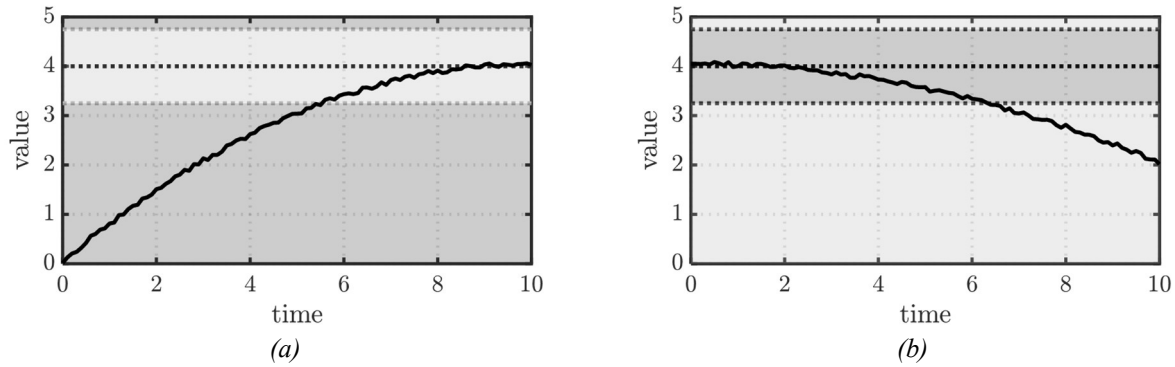
Command	Type	Description
release_stage	*Boolean*	True when conditions for stage separation are met.
engine_valve	*Boolean*	High state (True) when the propulsion system must be active and low state (False) when the engines must be off. Mimics the behaviour of the propulsion controlling valves.
release_fairing	*Boolean*	True when time for fairing release is reached.
eject_payload	*Boolean*	True when the guidance algorithm detects the arrival at target orbit.
mission_abort	*Boolean*	True when the safety block detects a degraded behaviour and triggers a flight termination alert. False when safety states nominal conditions.
switch_off	*Boolean*	True when guidance detects that the last last manoeuvre has been executed and thus the mission is completed.

2.2.2 Intelligent state transitions

The new approach to transitions presented in this paper allows for a certain level of flexibility to combat uncertainties or unpredictable perturbations during the flight. With this purpose, temporal based transitions are evolved to account for non-temporal variables that have the capacity to tune the transitions activation time to ensure that its values are within an optimal range to safely allow the transition. To simplify the design, the considered transitions explained hereafter are limited to bivariate decisions.

1. **Propulsion ignition:** $\boldsymbol{f = (t, \omega)}$. To allow engine ignitions, aside from time, the angular velocity (ω) is considered. By accounting the angular velocity in the authorization of the ignite command, the system is expected to also work as a safety measure. In that manner, the transition will not be allowed if the vehicle is under an uncontrolled tumbling (pitch and yaw axes). The spin (roll axis) is not accounted since it does not contribute to an uncontrol of the vehicle's attitude and thus it is not critical to inhibit the ignition of the engines.
2. **Propulsion cut-off**: $\boldsymbol{f = (t, \dot{m})}$. To allow engine cut-offs, the propellant mass flow ($\dot{m}$) is used as the complementary variable. In nominal conditions there is a decay of propellant mass flow at the time of cut-off from the nominal flow down to zero. However, due to perturbations present during the flight, this mass flow might experience a prior progressive decay. Mass flow is therefore considered in order to avoid low performances on the propulsive system causing a divergence from the mission nominal trajectory.
3. **Stage release:** $\boldsymbol{f = (t, P_c)}$. It has been proven by other launcher providers and operators that the pressure at the combustion chamber (P_c) is a key indicator of the propulsive system for a proper and safe stage separation. Therefore, chamber pressure is used to control the stage release command and avoid a hazardous stage separation causing a failure that leads to an abruptly end of the mission.

The nominal behaviour of the considered variables during the time intervals prior to its respective transitions present the two possible scenarios depicted in Figure 5. Variables might approach a nominal transition zone [bright area in Figure 5a] such is the case for the angular velocity and the chamber pressure, or tend to leave the nominal operation zone [dark area in Figure 5b] as is the case for the mass flow. Consequently, it is of key importance to develop a model that can easily handle both scenarios, as is the case of the proposed model for the intelligent transitions explained below.



(a) *(b)*

Figure 5. Representation of potential transition scenarios with (a) variables approaching a nominal zone and (b) variables distancing from the nominal zone. Numerical values are not representative.

DOI: 10.13009/EUCASS2022-4841

While in the discrete domain the presence of the variable of interest in the brighter and darker regions depicted in Figure 5 directly assesses a discrete yes or no decision to allow or block a certain transition, the new approach to transitions consider the optimality degree or closeness to the nominal case to allow or block the transition thanks to a semi-fuzzification of the system. Input variables are fuzzified using a membership function that associates the input to a degree of optimality to allow the transition, converting the discrete yes/no tags into a continuous tag indicating how optimal is the value with respect to the ideal transition.
Nonetheless, since the timeline of the mission is computed after optimizing the flight, temporal variables must have a higher importance on the allowance of the transitions. To that aim, the weighted mean of the fitness of each variable (f_i) is computed to find the global transition fitness (f) [see Eq.(5)]. Hence, by the appropriate choice of the weights for each variable (w_i), transitions can be modelled to compensate for small unpredicted perturbances.

$$\boldsymbol{f = \frac{\sum_i^N w_i \cdot f_i}{\sum_i^N w_i}} \tag{5}$$

Individual fitness for each variable is directly the degree of membership (dom) of the fuzzified variable. Therefore, it is crucial to correctly select the membership function to obtain the correct behaviour of the system.
Among the commonly used memberships, generalized bell functions defined by Eq.(6) is chosen. For the sake of simplicity, Eq. (6) is reduced by fixing $b = 1$, in order to become fully tuneable by two characterizing parameters $f(x, a, c)$, with a representing the aperture of the bell shape and c its centre value.

$$\boldsymbol{bell(x, a, b, c) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}}} \tag{6}$$

The choice of the bell function as membership is supported by its capacity to be implemented to the transitions of both scenarios described previously, with variables entering or leaving the optimal value. The fitness computation for the mass flow [behaviour from Figure 5b] is hence simply adapted to $f = 1 - bell(x, a, c)$. As can be observed in Figure 6, the crossing point between the two represented functions coincides with inputs at $x = c \pm a$. However, to avoid the transition allowance with one of the variables outside of the sigma region, the bell function is modified as depicted in Figure 7, assigning a zero degree of membership for all $x < c - a$ and $x > c + a$.

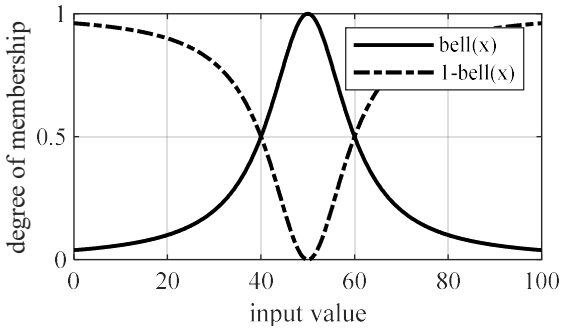


Figure 6. Membership functions for bell-shaped f(x,a,c)=bell(x,a,c) and its reflection g(x,a,c)=1-bell(x,a,c) for values of c=50 and a=10. Axes are dimensionless.

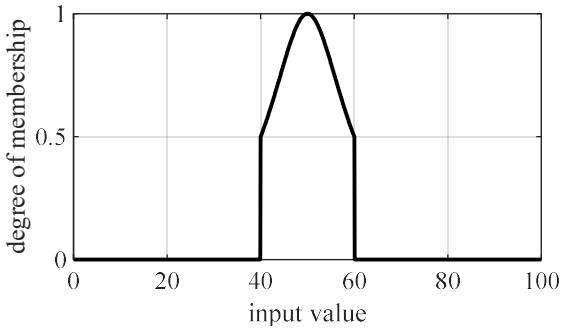


Figure 7. Adapted membership function with bell(x,a,c) for $c - a \leq x \leq c + a$ and 0 otherwise for values of c=50 and a=10. Axes are dimensionless.

Furthermore, since the loosening of the transition conditions would lead to a displacement of the transition allowance even in absence of perturbations, the temporal membership function must be adapted to avoid the displacement of the timeline events in the course of the nominal trajectory. Therefore, depending on the previously defined input behaviours [Figure 5] the time membership is nullified for all values previous of posterior to the nominal event time [see Figure 8].

DOI: 10.13009/EUCASS2022-4841

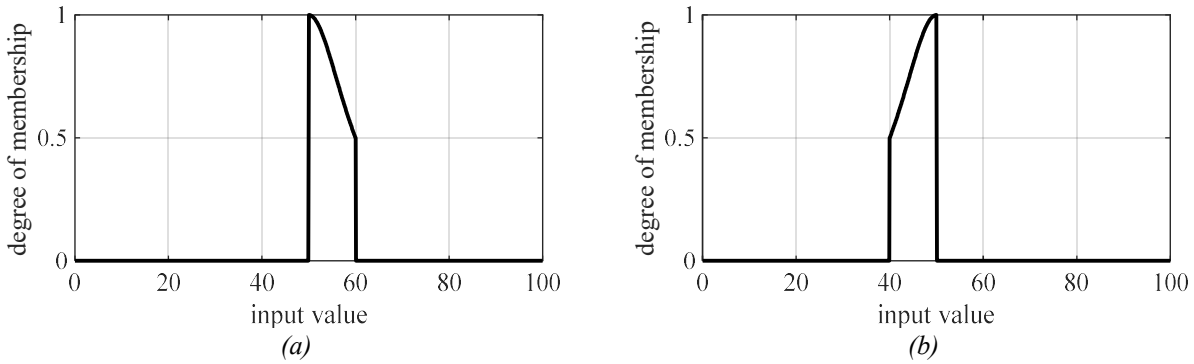


Figure 8. Asymmetric bell functions for the memberships of the temporal variables for allowing (a) time delays and (b) time advancements for values of c=50 and a=10. Axes are dimensionless.

The transition is then allowed only if the computed global fitness is equal or higher than a tuneable threshold in the range [0.5 0.9].
It is noteworthy that the intelligent transitions are only considered during the flight phase with open-loop guidance, since once the guidance is switched to closed-loop, it is the guidance algorithm itself who commands the propulsion system according to the orbital manoeuvres needs and hence, who triggers the transitions between the propulsion *on* and *off* states. Furthermore, the ignition of the main engine of the first stage is also not accounted to be intelligent since the vehicle is at the launch pad in known conditions and no flexibility shall be allowed.

3. Testing procedures

Testing processes are central to verify that the developed FSW behaves as expected and fulfils the requirements. This is translated into the need of a platform allowing great flexibility to simulate the software in multiple configurations to characterize the developed modules and find their operational range, all running in a realistic environment that emulates the mission conditions. A Software-In-the-Loop (SIL) platform [3] is used for this goal. The SIL approach constitute the base for further real testing through its progressive augmentation into Processor-In-the-Loop (PIL), by adding the real processor to run the FSW code in real-time while environment is simulated, and into Hardware-In-the-Loop (HWIL), where real hardware interfaces to the OBC are used.

3.1 Safety testing procedures

Testing procedures concerning the safety module are addressed to test the capacity of flexibility and configurability of the FLOS and to the assessment of the impact that the safety has in a degraded mission when integrated in the FSW.

3.1.1 Safety unit test

The testing procedures addressed to the safety are focused on the assessment of the flexibility and configurability of the FLOS bock. Hence, the tests are designed to obtain the variation in the percentage of positive termination decisions for an extensive set of FLOS configurations when exposed to a battery of inputs representing the whole spectrum of potential diagnosed scenarios, formed by the entire set of permutations of the arrays $[0\colon 0.05\colon 1]$ in each diagnosis module.
In terms of the set of configurations, four scenarios are envisioned as detailed in Table 8, allowing to assess the impact on the safety flexibility of tuning the membership functions or the rules weights.

Table 8. Description of the fixed and variable parameters of the configurations for the safety unit testing procedures.

ID	Fixed	Variable
rules@1	Rule weights [1 1 1 1 1 1] and membership function $a = 0$	All permutations of the three diagnosis for memberships $b = [0.01;\ 0.1\colon 0.1\colon 1]$
mfs@0.2	Memberships $a = 0, b = 0.2$	All permutations of the three set of rules for $w_i = [0.1\colon 0.1\colon 1]$
mfs@0.5	Memberships $a = 0, b = 0.5$	All permutations of the three set of rules for $w_i = [0.1\colon 0.1\colon 1]$
mfs@0.8	Memberships $a = 0, b = 0.8$	All permutations of the three set of rules for $w_i = [0.1\colon 0.1\colon 1]$

DOI: 10.13009/EUCASS2022-4841

3.2.1 Safety integration

Safety integration procedures are designed to test how the developed module integrates with the rest of the FSW in the simulation of a mission. However, since nominal trajectories are designed to not cause safety problems, a non-nominal trajectory must be created to provoke a safety violation scenario. The implemented degraded trajectory is an azimuthal deviation of 10º [see Figure 9] on the H2020 ENVOL nominal trajectory launching from Andøya, based on the anomaly occurred on the Ariane 5 flight V241 [6], causing the trajectory to cross the ILL and ensuring a dangerous situation where a terminate decision must be issued by the safety block.

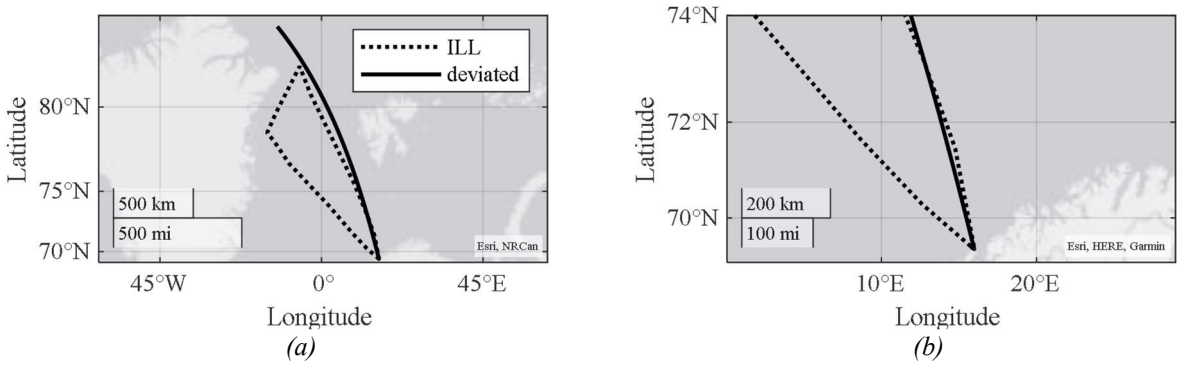


Figure 9. Mercator projection for the baseline ENVOL trajectory perturbed with an azimuthal deviation of 10º from t=H0. (a) General view of the first 500 s of trajectory and (b) zoom at the crossing point with the ILL.

To compensate the non-consideration of fragmentation models and the consequently reduced size of the computed impact areas through the covariance of the navigation state, a scale factor is added to the ellipses creation to simulate the effect of non-considered perturbations during the ballistic fall and evaluate the effect that the size of the computed impact area has on the safety decision flexibility capacity. The selection of the scaling factors is justified through the length of the semi-major axis of the resulting ellipse in a Mercator projection at latitude 70º, as seen in Table 9. Although the resulting impact area sizes might not be big enough to be fully representative of a real mission, due to the shape of the ILL higher scale factors cause a violation at the start of the simulation and cannot be tested.

Table 9. Dimensions of the impact area semimajor axis for a Mercator projection at λ=70ºN as function of the applied scale factor.

Scale Factor	Impact ellipse semi-major axis (Mercator projection @ $\lambda = 70º$)
1	≈ 90 m
10	≈ 300 m
10^2	≈ 900 m
10^3	≈ 3 km
10^4	≈ 9 km

These scale factors conform a test suite each, with their respective internal testcases formed through the tuning of the b-parameter of the impact point membership function for the values [0.01 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0].

3.2 Flight Manager testing procedures

Testing procedures for the flight manager are addressed to the characterization of the intelligent transitions via unitary testing and the assessment of the flags management along a nominal mission achieved with the integration of the flight manager into the FSW.

3.2.1 Flight Manager unit test

As mentioned, unit testing procedures for the flight manager are designed to analyse the capacity of flexibility allowed by the semi-fuzzification of the state-machine transitions as well to characterize its behaviour. To do so, a single transition of a variable approaching the optimal zone is evaluated in a configuration with $t = [95{:}\,0.01{:}\,105]$, $t_0 = 100$, $x_0 = 100$, $w_t = 1$, $w_x = 1$, $\sigma_t = 1$ and $\sigma_x = 0.05$.

The input approaches quadratically the nominal value following the expression $y = 0.1(t - t_0 - k_i)^2$ where t is the time of the mission, t_0 is the nominal transition time as specified in the mission timeline and k_i is an offset factor with values $k = [0\ 0.6\ 0.8\ 1.0\ 1.2\ 1.4\ 1.6\ 1.8]$ used to simulate a perturbation in the input value causing a delay on the

DOI: 10.13009/EUCASS2022-4841

achievement of the transition optimal conditions. Additionally, a second test case is designed to evaluate the impact of random noise in the input with the expression $y = 0.1(t - t_0 - k_i)^2 - 0.05 + 0.1rand()$, with a sample size of N=1000 cases to evaluate the mean and variance introduced by the randomness. In both cases, each input is evaluated for multiple decision allowing thresholds at $[0.5{:}\,0.05{:}\,0.9]$.

3.2.2 Flight Manager integration

Flight manager integration test procedures focus on its function as manager of the other FSW blocks and do not consider intelligent transitions but a configuration strictly obeying the mission timeline programmed. The testing is aimed to inspect the evolution of the flags sent to the OBC and the corresponding commands sent to the SVF platform. The mission implemented corresponds to the nominal of the H2020 ENVOL, targeting a sun-synchronous orbit at 600 km and with the timeline specified in Table 10.

Table 10. Timeline of the nominal major events programmed for the baseline trajectory of the project ENVOL.

Event	Time
Power-up	H0 – 60 s
First stage engines ignition	H0
First stage engines cut-off	H0 + 100 s
First stage release	H0 + 101 s
Second stage engines ignition	H0 + 104 s
Fairing release	H0 + 169 s
Second stage engines cut-off	H0 + 204 s
Second stage release	H0 + 205 s
Third stage engines ignition	H0 + 209 s
Third stage engines cut-off	H0 + 781 s
Circularisation manoeuvre start	H0 + 4438 s
Payload injection	H0 + 4447 s

4. Results

Figure 10 and Figure 11 show the results of the unit test procedures for the safety module, showing its capacity of configurability to adapt to multiple levels of restrictiveness. As can be observed in Figure 10, the tuning of the membership functions (test case rules@1) presents low skewed data with the median at 8% and the first and third quartiles at ≈2% and 20% respectively, leading to an interquartile range of ≈18%, and a high presence of distributed outliers in the high percentages range between ≈45% and ≈85%. The descending behaviour visible in Figure 11 subplot rules@1 is only consequence of the order in which configurations are created, from more restrictive to more tolerant. On the other hand, the tuning of the rule weights (test cases mf@02, mf@05 and mf@08) show a practically inexistent spread of the data with first and third quartiles coinciding with the median at ≈53%, ≈14.5% and ≈1.5% respectively. Figure 11 shows this high repeatability of results around the median value.

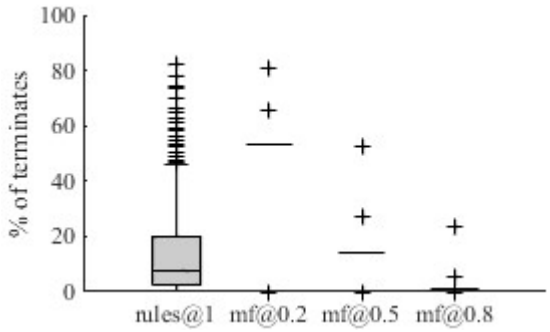


Figure 10. Distribution of the percentages of issued terminate commands as function of the tuning parameter.

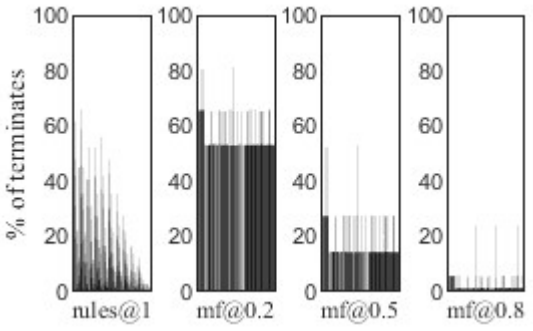


Figure 11. Percentage of issued terminate commands for the set of configurations achievable through the tuning of the membership function (rules@1) and the tuning of the rule weights (mf@02, mf@05 and mf@08).

DOI: 10.13009/EUCASS2022-4841

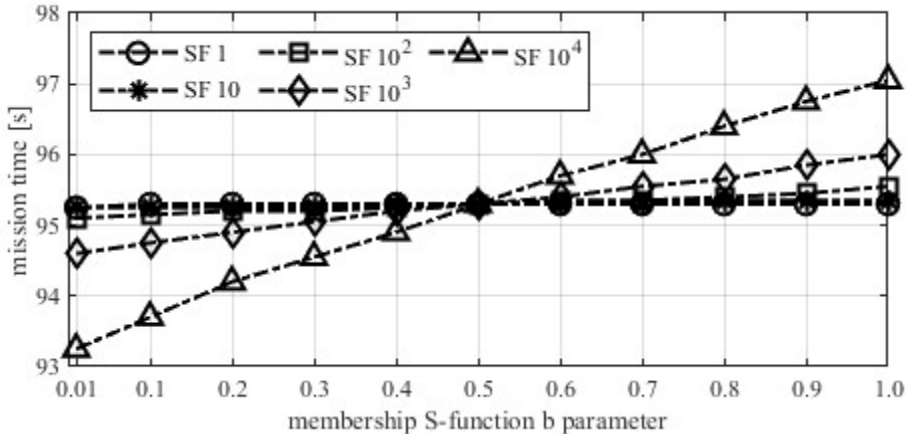


Figure 12. Mission abort time as function of the restrictiveness of the FLOS (b parameter) and the impact area dimensions (SF) for the ENVOL baseline trajectory with an azimuthal deviation of 10º.

Flight manager unit testing results on the intelligent transitions are summarized in *Table 11* for the noiseless scenario and in *Table 12* for the noisy test case. As can be observed from the obtained values, the increment of the threshold value causes higher transition delays for a given perturbation level up to the maximum of 1 s defined by the configured σ_t. Furthermore, the blocking of transitions (represented as empty table cells) due to not enough optimality increase with the increase of the perturbation offset *k*. The same behaviour is obtained on the noisy scenario but with a slight deviation between the mean delays and the corresponding nominal input delays that cause differences in allowing or blocking the transition for the small perturbations in the most restrictive thresholds.

Table 11. Transition delays in seconds for a noiseless quadratic input with abscissa offset k.

	Threshold								
	0.50	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**
k=0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
k=0.6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0300	0.1100
k=0.8	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1900	0.2800	-
k=1.0	0.3000	0.3000	0.3000	0.3000	0.3000	0.3800	0.5100	-	-
k=1.2	0.5000	0.5000	0.5000	0.5000	0.6100	0.7600	-	-	-
k=1.4	0.7000	0.7000	0.7300	0.8400	0.9900	-	-	-	-
k=1.6	0.9000	0.9500	-	-	-	-	-	-	-
k=1.8	-	-	-	-	-	-	-	-	-

In terms of results for the integration of the safety module, Figure 12 depicts the time at which the deviated mission is terminated as function of the impact point membership function b value for multiple scale factors. As can be extracted from the slope of the results for each scale factor, the higher this factor is, thus the larger the computed impact area, the higher the capacity to advance or delay the terminate command and therefore the higher the flexibility achievable by the FLOS. It is noteworthy to notice that for a configuration with $b = 0.5$ the time of terminate is independent of the scale factor.

DOI: 10.13009/EUCASS2022-4841

Table 12. Transition delays in seconds expressed as mean ± one standard deviation for a quadratic input with abscissa offset k and white noise of amplitude 0.1. Statistics obtained with a sample of size N=1000.

	Threshold								
	0.50	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**
k=0	0.0000 ±0.0000	0.0000 ±0.0000	0.0000 ±0.0000	0.0000 ±0.0000	0.0000 ±0.0000	0.0000 ±0.0000	0.0019 ±0.009	0.0034 ±0.0047	0.0050 ±0.0050
k=0.6	0.0054 ±0.0090	0.0054 ±0.0090	0.0054 ±0.0090	0.0054 ±0.0090	0.0054 ±0.0090	0.0054 ±0.0090	0.0083 ±0.0122	0.0111 ±0.0145	0.0142 ±0.0174
k=0.8	0.0164 ±0.0194	0.0164 ±0.0194	0.0164 ±0.0194	0.0164 ±0.0194	0.0164 ±0.0194	0.0164 ±0.0194	0.0238 ±0.0249	0.0324 ±0.0302	0.0483 ±0.0404
k=1.0	0.0872 ±0.0451	0.0872 ±0.0451	0.0872 ±0.0451	0.0872 ±0.0451	0.0872 ±0.0451	0.0894 ±0.0475	0.1415 ±0.0518	0.1924 ±0.0531	0.2507 ±0.0576
k=1.2	0.2859 ±0.0454	0.2859 ±0.0454	0.2859 ±0.0454	0.2859 ±0.0454	0.2861 ±0.0459	0.3229 ±0.0547	0.3851 ±0.0579	0.4487 ±0.0590	0.4629 ±0.0205
k=1.4	0.4862 ±0.0455	0.4862 ±0.0455	0.4862 ±0.0455	0.4871 ±0.0476	0.5316 ±0.0551	0.5969 ±0.0565	0.6611 ±0.0564	-	-
k=1.6	0.6886 ±0.0454	0.6886 ±0.0454	0.6930 ±0.0510	0.7527 ±0.0552	0.8144 ±0.0568	0.8746 ±0.0502	-	-	-
k=1.8	0.8853 ±0.0421	0.8931 ±0.0460	0.9397 ±0.0353	0.9714 ±0.0215	0.9939 ±0.0070	-	-	-	-

Figure 13 and Figure 14 show respectively the evolution of the flags sent by the flight manager to the OBC for FSW configuration and the commands sent to the SVF to manage the behaviour of the launch vehicle for the whole baseline mission ending with the injection of the payload at the targeted orbit.

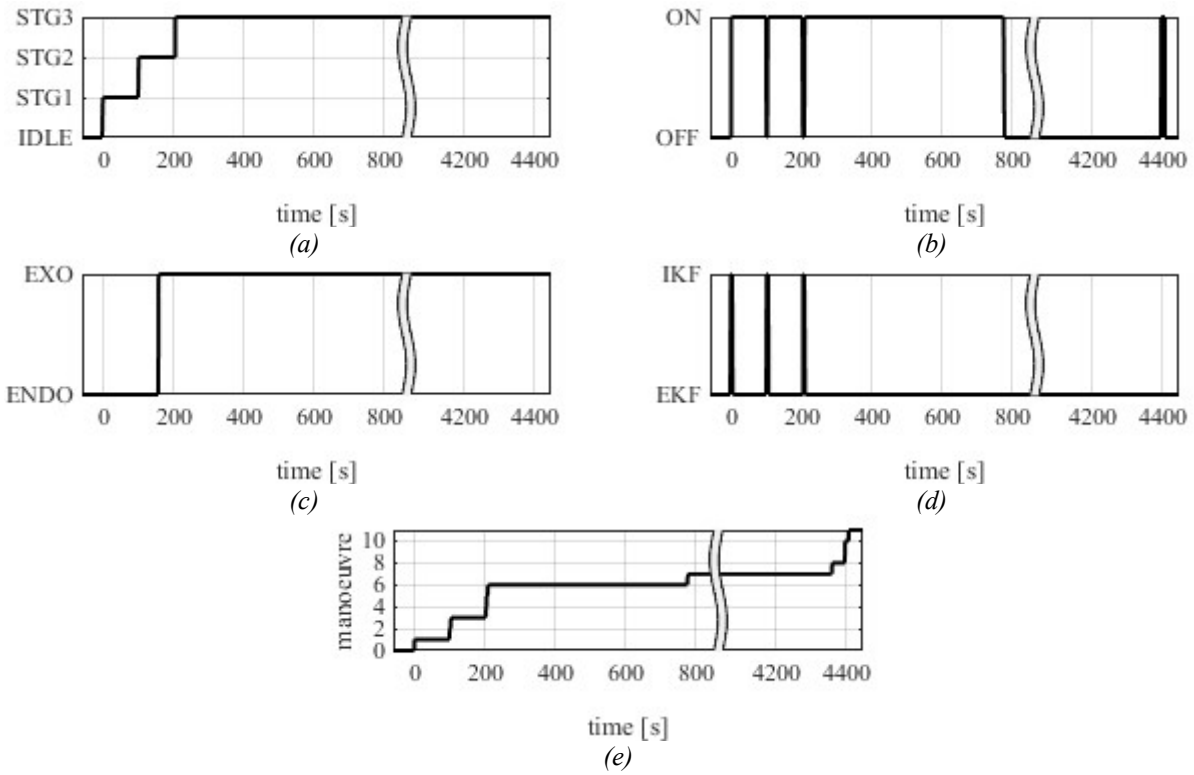


Figure 13. Evolution of the flags along the baseline trajectory for the ENVOL. (a) operative stage, (b) propulsion status, (c) atmospheric conditions, (d) Kalman filter and (e) manoeuvre counter.

DOI: 10.13009/EUCASS2022-4841

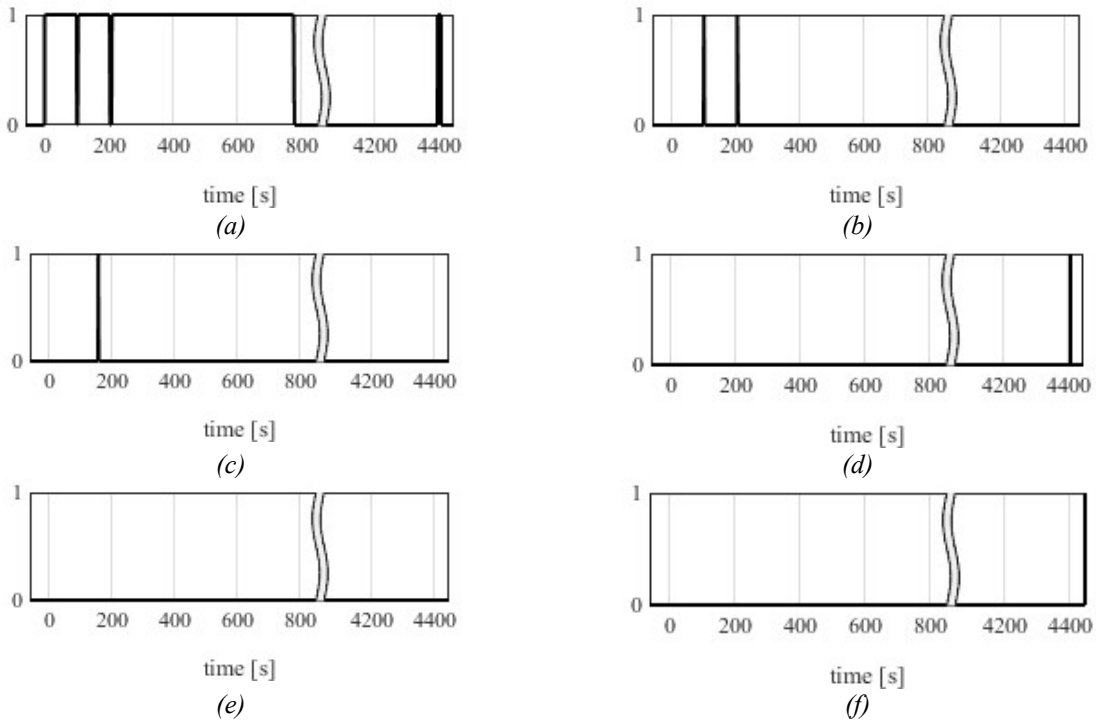


Figure 14. Evolution of the commands along the baseline trajectory for the ENVOL. (a) engines ignition, (b) stage release, (c) fairing release, (d) payload injection, (e) mission abort and (f) FSW switch-off.

5. Discussion

The FLOS module is shown to be configurable and flexible to adapt to multiple restrictiveness requirements. The reached decisions are seen to be highly conditioned by the membership function, allowing a high configurability through its tuning, while the flexibility obtained through the modification of the rule weights is limited. However, this weighing has the potential to be used as a switch to turn off diagnosis modules during certain flight phases such as the impact predictor when the vehicle reaches satelization conditions. This enhances the adaptability of the system, allowing it to be missionisable and reduce the duration of mission campaigns thanks to a lower validation requirement. Regarding its repercussion on the mission abort for a deviated trajectory, the obtained data shows the influence of the magnitude of the impact area on the flexibility of the module, or equivalently, on the variability of the time of termination as function of the restrictiveness of the impact area in conflict. With the current implementation for the computation of the impact ellipse, that is with unit scale factor, the area is too small to take advantage of the module's flexibility since the distance between two consecutive navigation points is too large compared with the ellipse semimajor axis and as consequence their areas do not interest, nullifying the flexibility of the FLOS. Furthermore, for a configuration allowing half the area to be in conflict ($b = 0.5$) the system is independent of the scale factor and behaves as a classical system only considering whereas the point of impact is inside or outside the protected area. This behaviour is consequence of considering the impact areas as ellipses since for all convex polygons more than the 50% of the area will be outside the protected area only when its centre is also outside.

Analogously, flight manager intelligent transitions are shown to provide flexibility against perturbations but be highly dependent on the behaviour of the input variable and on the authorization threshold, capable of tuning the restrictiveness of the transition. The higher the threshold, the closer the non-temporal shall be to its optimal conditions as time moves away from its nominal and therefore, the more restrictive the system is. The restrictiveness of high thresholds is seen through the denegation of transitions when the input is non-nominal, thus providing less capacity to absorb perturbations and causing the abortion of the mission in degraded scenarios. On the other hand, a lower threshold is translated into more permissiveness on the state of the non-temporal variable, allowing the mission to continue but in less optimal conditions. Notwithstanding, integration tests are required to check which is the effect on the payload injection conditions to evaluate if this relaxation enhances mission success, leading to a higher robustness of the FSW against degraded performances of the vehicle and external perturbations.

Moreover, as seen in the results section the effect of a noisy input is translated into having an uncontrolled variability at the time of transition authorization. Despite this noise effect must be minimized through the correct choice of the transition optimal range (σ_x), a debouncing system shall also be implemented to avoid transitions caused by peaks of

DOI: 10.13009/EUCASS2022-4841

membership consequence of the noise. This system to be developed shall wait for a number of consecutive allowances before authorizing the transition.
While the test is not conclusive due to a lack of representativeness, it serves as an approximation to study the effect of the flexibility on the transitions for a common input behaviour among the considered variables. Nonetheless, the results open the possibility to use this flexibilization as a pseudo-diagnosis for safety by blocking the events in case of undesired conditions, e.g. uncontrolled tumbling between the stage release and the next stage engines ignition.
Lastly, regarding the management capacity of the flight manager, the system is proved to be capable of correctly follow the programmed timeline and configure the FSW to successfully achieve the target orbit in satisfactory conditions to inject the payload.

6. Conclusions

As the satellite industry advances towards the NewSpace concept, it is important to understand how the development of innovative solutions in the FSW affect its drivers in terms of missionisation, reusability, reliability and cost effectiveness. The final objective is to have a reliable and configurable on-board AFTS based on the modules presented in the current paper, that would contribute to the NewSpace requirements and operations. The work proves the configurability of the presented safety module through the tuning of the membership functions fuzzifying the health status obtained through the diagnosis blocks. Although having a low impact on the decision flexibility the weighting of the rules serves as a switch-off to activate or deactivate diagnostics as the mission requires for each flight phase, providing adaptability to the system and thus contributing to the decrease of mission campaign duration. Nonetheless, flexibility of the system is closely attached to the dimensions of the computed impact area since it is the driver of the clipping algorithm that will compute the percentage of area in threat. As consequence of the impact areas being convex polygons, with a configuration that considers danger for areas in hazard of ≥50% of the impact area the system is capable of acting as if only considering the impact point, as traditionally. This allows to address the multiple roads of development planned by spaceport operators in the New Space domain. Moreover, the management of the FSW configuration by means of a state machine has been proven to be accurate and effective with the added advantage of providing the opportunity to exploit the transitions. The semi-fuzzification of the interstate transitions demonstrated the capacity of adjustment of the system restrictiveness through the tuning of the authorization threshold and showed the necessity of the development of a debouncing system to combat noisy inputs. Although further testing to assess its impact on mission success is required, the intelligent transitions are found to be favourable to be used as pseudo-safety thanks to its capacity of blocking the mission major events if in degraded conditions.
Future research into OASSYS should therefore focus on the study of real-time algorithms to compute the impact area of the launcher based on fragmentation models that is dependent on the navigation state of the vehicle (altitude from ground, velocity, attitude), as well as on the development of the diagnosis modules for the flight corridor and the IVHM. Furthermore, the research into the flight manager should point to the development of the debouncing system to combat noisy inputs and to the integration and evaluation of the intelligent transitions advantages and drawbacks on mission objectives. The test of the present work would fulfilled and its performance assessed, once integrated with the rest of the FSW modules. The tool developed at GTD [3] supports further and detailed studies on the AFTS performance towards mission safety analysis validating the system for operational flights.

References

[1] Wekerle, T., Filho, J. B. P., Costa, L. E. and Trabasso, L. G., 2017. Status and trends of smallsats and their launch vehicles – an up-to-date review. *Journal of Aerospace Technology and Management* **9(3)**, 269-286.
[2] Williams, C., Doncaster, B., and Shulman, J., 2018. Nano/Microsatellite market forecast 8th edition. SpaceWorks Enterprises, Inc. (SEI).
[3] Sabán, A., Martín, J., Díez, E. and Martín, N., To be published. 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).*
[4] Vallverdú, D., Pou, C., Badenas, M. and Díez, E., 2018. A Real-Time Hybrid Navigation System for an Autonomous Space Launch Vehicle. In *9th Embedded Real Time Software and Systems (ERTS) European Congress.*
[5] Vallverdú, D., Pou, C. and Díez, E., 2019. Design of an on-board flight safety system for space microlaunch vehicles. In *8th European Conference for Aeronautics and Space Sciences (EUCASS).*
[6] Arianespace, 2018. Independent Enquiry Commission announces conclusions concerning the launcher trajectory deviation during Flight VA241. Accessed 15 March 2022, <https://www.arianespace.com/press-release/independent-enquiry-commission-announces-conclusions-concerning-the-launcher-trajectory-deviation-during-flight-va241/>