doi: 10.82124/CEAS-GNC-2026-090 CEAS-GNC-2026-090

Safety Filter for Rocket Thrust Vector Control: Monitoring and Enforcing Flight-Envelope Constraints

Aitor R. Gomez Postdoc, Section of Automation & Control, Aalborg University, Aalborg, Denmark. arg@es.aau.dk

João Belfo GNC Engineer, DEIMOS Engenharia SA, GNC/AOCS Competence Center, 1070-061, Lisbon, Portugal. jpbelfo@indracompany.com

Paulo Rosa Head of Flight Segment, DEIMOS Engenharia SA, GNC/AOCS Competence Center, 1070-061, Lisbon, Portugal. parosa@indracompany.com

Pedro Simplício GNC Engineer, European Space Agency, ESTEC, Noordwijk-Binnen, The Netherlands. pedro.simplicio@esa.int

Jakob Stoustrup Professor, Section of Automation & Control, Aalborg University, Aalborg, Denmark. jakob@es.aau.dk

ABSTRACT

This paper presents a control strategy that brings the monitoring and enforcement of flight-envelope constraints in launch vehicles down to the GNC level. This technology, widely known as safety filters in robotics and the automotive sector, serves two main purposes: to provide a formal methodology of addressing safety-critical state constraints, and to remain modular and minimally invasive to existing GNC architectures. Yet, many additional advantages stem from the integration of safety filters in launchers. They offer a substantial reduction in validation and verification, and missionization efforts; they ensure increased launch opportunities under stronger weather conditions, and they allow a redistribution of efforts between computational guidance and control. The safety filter proposed in this paper is based on a robust version of high-order control barrier functions, which are formulated as efficient, quadratic optimization problems that account for unknown, but bounded, wind disturbances. Moreover, it produces residual signals that carry essential information for safety monitoring, autonomous flight termination capabilities and guidance re-computation algorithms. We also present a relevant tradeoff between stability and safety manifested in launchers and other unstable systems, which highly influences the design of the safety filter. Finally, we demonstrate the effectiveness of this technology in a Monte Carlo campaign using a high-fidelity simulator.

Keywords: Safety filter, Launch vehicles, Flight corridor, Optimization, Robustness

Nomenclature

CBF	=	Control Barrier Function
GNC	=	Guidance, Navigation and Control
HOCBF	=	High-Order CBF
LTI	=	Linear time-invariant
TVC	=	Thrust Vector Control
V&V	=	Validation and Verification
$\mathcal{C}, \partial\mathcal{C}, \text{Int}\mathcal{C}$	=	Set of safe states, boundary and interior
G_L, Δ_L	=	Launcher linear model and unstructured uncertainty
G_{TVC}, Δ_{TVC}	=	TVC dynamics and unstructured uncertainty
G_τ, Δ_τ	=	TVC delay model and unstructured uncertainty
G_w	=	Wind disturbance model
K	=	Nominal controller in frequency domain
$L_f h$	=	Lie derivative of h along f
Q	=	Dynamic pressure
$W_{(\cdot)}$	=	Control requirements
$\mathcal{U}$	=	Set of admissible control inputs
$\mathcal{W}$	=	Set of admissible disturbance inputs
$\mathcal{X}$	=	Set of admissible states
a, b	=	Lower and upper bounds on the lateral drift
f, g, q	=	Dynamics, input and disturbance functions
$h_{(\cdot)}$	=	Safety-critical constraint or CBF
k	=	Nominal control feedback in time domain
u_ψ, u_θ	=	Nominal control input in yaw and pitch channels
u_ψ^*, u_θ^*	=	Safe control input in yaw and pitch channels
w_ψ, w_θ	=	Disturbance input vector in yaw and pitch channels
w_∞	=	Infinity norm of wind disturbance signal
x	=	State vector in yaw channel
y, z	=	Lateral drift states
α	=	Angle of attack
$\alpha_i(\cdot), \beta_i(\cdot)$	=	Class-$\mathcal{K}$ functions
δ	=	Slack variable
φ_i	=	High-order CBF conditions
ψ, θ	=	Yaw and pitch angles
K_a, ω_a, ξ_a	=	TVC model parameters
τ	=	TVC delay
γ_1, γ_2	=	Maximum lateral drift and attitude error

1 Introduction

Safety, stability, and performance are three critical aspects of any dynamical system. For decades, however, stability and performance have been the primary focus of Guidance, Navigation, and Control (GNC) algorithm design [1, 2], as well as of Validation and Verification (V&V) activities [3, 4], for launch vehicles. This emphasis has often relegated safety to a feature addressed only at the highest levels of system functionality, commonly managed by passive methods or by a human-in-the-loop responsible for overseeing and enforcing safety during flight. This situation reveals two important factors. First, only highly trained personnel are typically capable of assessing pre-flight conditions and monitoring launcher's health and safety during flight. Second, rapid decision-making becomes critical when safety

is compromised. Under this prospect, unforeseen flight conditions, poor decision-making, and delays are human limitations that can easily pose launch activities at risk. To protect launchers against these elements, it is essential to reduce reliance on human intervention and instead incorporate autonomous capabilities that can monitor and ensure safe operations during flight.

This paper is concerned with improving one particular aspect of safety: preventing the launcher from leaving the flight corridor. Indeed, extending the control loop to achieve this task autonomously can significantly enhance the reliability of the launcher while also offering other equally important benefits. For instance, it allows launch vehicles to operate under stronger wind conditions while ensuring they remain within the designated operating region, thereby expanding launch opportunities and increasing scheduling flexibility. These benefits address some of the limitations of current GNC architectures, which primarily focus on stability and performance, and often lack mechanisms to ensure the system remains within constrained operational boundaries. As a result, launch opportunities are currently restricted by the environmental conditions that existing GNC systems can tolerate.

Moreover, computational guidance algorithms [5] have been adopted by NASA and SpaceX to address complex task, such as the soft-landing problem. Lossless [6] and successive convexification [7] were developed and employed to effectively find feasible solutions to increasingly complex optimal control problems, and have now become widespread. While they have been proven useful for real-time trajectory planning, they still require high computational overhead, extensive validation and verification, and are not as reliable against uncertain environments as dedicated control strategies. The control extension presented in this paper, known as *safety filter*, enables transferring more authority to the control loop, allowing simplifications to the guidance problem, and resulting in improved computational efficiency and responsiveness to dynamic conditions. These simplifications also contribute to reduced validation and verification efforts associated with both, the guidance and control systems.

In recent years, safety filters have been matured to provide mathematical guarantees of *safety*, a term which in the context of autonomous dynamical systems takes on a very concrete meaning that enables the use of mathematical formalisms. These same formalisms are at the core of this work. To that end, safety filters can be designed using multiple approaches, each of which yield different levels of conservativeness, computational load and control intervention [8–10]. Furthermore, safety filters offer three advantageous features that make them very appealing for launchers: 1) they provide mathematical guarantees, reducing subsequent V&V efforts, 2) they are modular, and thus compatible with existing GNC architectures, and 3) they modify the controller input minimally, and only if safety is compromised. Due to these strengths, safety filters have been extensively used in Robotics (specially for collision avoidance [8, 11]) and are now beginning to emerge in the space sector for tasks like rendezvous and docking operations for satellites [12, 13] or satellite attitude control [14]. The transition to the space transportation is a natural next step.

The safety filter developed in this work is based on a robust Control Barrier Function (CBF) formulation that aims to (minimally) modify the launcher's thrust vector control input so as to ensure it remains within the flight corridor. This is done by finding a control input that satisfies certain safety conditions given by the CBFs, while minimizing the distance to a nominal control input provided by a robust H_∞ controller. The resulting control input theoretically guarantees satisfaction of safety-critical constraints, while simultaneously providing the stability and performance specifications of the robust nominal controller when safety is not compromised.

In practice, these guarantees are harder to obtain simultaneously due to a key characteristic observed in safety filters, i.e. they may occasionally compromise the system's stability in order to maintain or improve safety. This fact indeed leads to a stability-safety tradeoff that can be omitted in many systems, but not in launchers and systems alike, as we will explain. We resolve this competition by relaxing the safety conditions, which will be derived in this paper, thus prioritizing stability. In the sequel, we provide further insight about how this relaxation can be leveraged to obtain a valuable signal for safety monitoring.

The choice of employing a CBF-based safety filter is motivated by the stringent computational limitations present on launch vehicles. Therefore, a certain degree of conservativeness is expected from this technique with respect to other safety-critical control strategies, in exchange for efficiency and fast computation. The safety conditions are expressed in terms of the lateral drift states, requiring High-Order Control Barrier Functions (HOCBFs) [15], which are extended to improve robustness against uknown, but bounded, wind disturbance based on the approach presented in [16].

The remainder of the paper is organized as follows. Section 2 is a brief introduction to safety-critical control, presenting formal definitions of safety and relevant results from CBF literature employed throughout the paper. In Section 3, we introduce the overall control architecture, describe briefly the design of the nominal controller and develop the launcher safety filter. In addition, we present and address the effects of the stability-safety tradeoff. Finally, Sections 4 and 5 discuss the high-fidelity simulation results and conclude the paper, respectively. We include an Appendix section with the linearized models employed in the control design and other alternatives.

2 Preliminaries

A review of safety and CBF-based safety filters is provided in this section, since the definitions and results introduced here will be instrumental in the remaining of the paper. The reader is referred to [8] for a more detailed explanation of these concepts. We start by defining the notion of safety, and how it can be certified mathematically on dynamical systems. Then, we introduce the definition of high-order CBF (HOCBF), which allows us to classify and synthesize safe control actions. Lastly, we show how the safety filter formulation leverages HOCBFs to redesign a control input from a nominal controller to make it safe. To that end, consider the following input-affine nonlinear system:

$$\dot{x} = f(x) + g(x)u + q(x)w, \tag{1}$$

where $x \in \mathcal{X} \subset \mathrm{R}^n$ is the state vector, $u \in \mathcal{U} \subset \mathrm{R}^m$ is the control input, $w \in \mathcal{W} \subset \mathrm{R}^d$ is the disturbance, and $f : \mathcal{X} \to \mathrm{R}^n$, $g : \mathcal{U} \to \mathrm{R}^{n\times m}$ and $q : \mathcal{W} \to \mathrm{R}^{n\times d}$ are Lipschitz continuous and differentiable maps. The affine structure in (1) appears in many mechanical systems, including those in robotics, automotive and aerospace.

2.1 Safety

The concept of safety in deterministic dynamical systems pertains to the mathematical property of *forward set invariance* with respect to a set defined by state constraints. It should be distinguished from conventional interpretations of safety, which often relate to physical hazards or human risk. Here, safety is defined in terms of system trajectories remaining within prescribed sets over time. For now, we will assume that no disturbances are present, thus $w = 0$ until otherwise specified.

We can begin to formalize a mathematical notion of safety by specifying a continuously differentiable function $h : \mathrm{R}^n \to \mathrm{R}$, referred to as *barrier* function, which characterizes a collection of safe states by means of its zero super-level set, denoted C.

$$C := \{x \in \mathcal{X} : h(x) \geq 0\}, \tag{2}$$

$$\partial C := \{x \in \mathcal{X} : h(x) = 0\}, \tag{3}$$

$$\mathrm{Int}C := \{x \in \mathcal{X} : h(x) > 0\}. \tag{4}$$

Herein, ∂C and Int$C$ denote the boundary and interior of the safe set, respectively. Forward set invariance implies that any trajectory of the system that departs from C, will always remain in C. Formally, this means that safety is achieved if, for a given feedback law $u = k(x)$, the following forward invariant

condition holds true.

$$\forall x(0) \in C \Rightarrow x(t) \in C,\ \forall t > 0.$$

Note that the previous condition involve two elements: the safe set C, defined by the function h, and the trajectories of the system, which are governed by the functions f and g. It is therefore expected that the necessary and sufficient conditions that verify forward invariance of (1) with respect to C depends on all three functions. Such *barrier* condition is given in [8] for compact[1] sets C.

$$C \text{ is invariant} \quad \Leftrightarrow \quad \dot{h}(x,u) \geq -\alpha_0(h(x)),\ \forall x \in C, \tag{5}$$

where $\alpha_0 : (-a, b) \rightarrow (-\infty, \infty)$, with $a, b > 0$, is an extended class-$\mathcal{K}$ function, i.e. continuous, monotonically increasing and $\alpha_0(0) = 0$. The dependency on the functions f and g is implicit, and just becomes apparent when differentiating h.

$$\dot{h}(x,u) = L_f h(x) + L_g h(x) u, \tag{6}$$

where $L_f h(x) = \frac{dh}{dx} f(x)$ and $L_g h(x) = \frac{dh}{dx} g(x)$ are lie derivatives.

Remark 2.1: *Note that if the system is at the boundary of the safe set, $x \in \partial C$, the barrier condition becomes $\dot{h}(x) \geq 0$ which implies the system will either remain on the boundary ($\dot{h} = 0$) or evolve towards the interior of C ($\dot{h} > 0$). This clear interpretation of why trajectories cannot leave the set C stems from Nagumo's theorem [17].*

Remark 2.2: *The barrier function h can be seen as a generalization of Lyapunov functions for sets, and as such, shares many properties with Lyapunov theory. In fact, it can be shown that a barrier function that renders C forward invariant also induces a Lyapunov function that renders C asymptotically stable [18].*

It occurs in our problem, as it is often the case in mechanical systems, that the term $L_g h(x) = 0$ for all $x \in C$, preventing the control u from explicitly appearing in condition (5). This typically arises when the safety constraint $h(x) \geq 0$ is defined on state variables that are not directly actuated but are instead separated from the control input u by one or more integrators (e.g. position or attitude constraints). In these cases, it is said that the barrier function (if it exists) is of relative degree two or higher, respectively. A well-established method to handle such constraints is by defining the following series of higher-order barrier functions, as shown in [15]:

$$\begin{aligned}
\varphi_0(x) &:= h(x) \\
\varphi_1(x) &:= \dot{\varphi}_0(x) + \alpha_1(\varphi_0(x)) \\
\varphi_2(x) &:= \dot{\varphi}_1(x) + \alpha_2(\varphi_1(x)) \\
&\vdots \\
\varphi_r(x,u) &:= \dot{\varphi}_{r-1}(x,u) + \alpha_r(\varphi_{r-1}(x))
\end{aligned} \tag{7}$$

where $\alpha_1, ..., \alpha_r$ are class-$\mathcal{K}$ functions. Then, the system (1) with feedback $k(x)$ is forward invariant with respect to the set $C_\cap = C \cap C_1 \cap C_2 \cap \cdots \cap C_r$ if

$$\varphi_r(x, k(x)) \geq 0,$$

where $C_1 := \{x \in \mathcal{X} : \varphi_1(x) \geq 0\}, \ldots, C_r := \{x \in \mathcal{X} : \varphi_r(x, k(x)) \geq 0\}$.

[1] A set is compact if its level sets are closed and bounded.

2.2 Robust Control Barrier Functions

A key advantage of this formulation is the ability to synthesize a control input u that enforces the barrier condition $\varphi_r(x,u) \geq 0$ at every point of the trajectory, rendering the set $C_\cap$ invariant and therefore safe. Thus, assuming that no feedback law is now available, the following set of safe control actions at a point x can be constructed:

$$\mathcal{U}_s(x) := \{u \in \mathcal{U} : \varphi_r(x,u) \geq 0)\}, \tag{8}$$

Then, a continuous and r-th order differentiable function $h : \mathcal{X} \rightarrow \mathrm{R}$ is a high-order control barrier function (HOCBF) for the system (1) if there exist class-$\mathcal{K}$ functions $\alpha_1, ..., \alpha_r$ such that the set $\mathcal{U}_s(x)$ is non-empty. If the set is empty, then there are no (safe) inputs that can guarantee $h(x) \geq 0$.

Remark 2.3: *We are interested in the previous definition since the main goal is to find control actions that bound the growth in lateral drift states, which have relative degree two with respect to the actuation u. Moreover, the safety conditions can be extended to the time-varying case without loss of generality. Then, the constraint $h(x,t)$, and also functions $\varphi_0(x,t), ..., \varphi_r(x,u,t)$, may additionally depend on t.*

Let us assume now that $w \neq 0$, and consider the major disturbance encountered in the launcher's atmospheric phase: wind. If unaccounted, wind can push the system's states outside of the safe set C. In order to make the HOCBF robust against this disturbance, we assume that the disturbance is unknown but bounded by the ∞-norm, $w(t) \leq w_\infty = ||w||_\infty$, for any $t > 0$. Then, following [16], we introduce a buffering term to the HOCBF condition in (8) that shrinks the safe set $C_\cap$ by an amount relative to the size of w_∞. This prevents the system from leaving the original set $C_\cap$ (and also C, given that it departs from $C_\cap$), even in the presence of the worst disturbance. Assuming that the relative degree of h with respect to the input u and the disturbance w are the same, the robust HOCBF becomes:

$$\varphi'_r(x,u) = L_f\varphi_{r-1}(x) + L_g\varphi_{r-1}(x)u + ||L_q\varphi_{r-1}(x)||w_\infty + \alpha_r(\varphi_{r-1}(x)), \tag{9}$$

where $L_q h(x) = \frac{dh}{dx}q(x)$. Now, the set of control actions that keep the system robustly safe at x is then:

$$\mathcal{U}'_s(x) := \{u \in \mathcal{U} : \varphi'_r(x,u) \geq 0)\}. \tag{10}$$

2.3 Safety Filter Formulation

Our aim is to use robust HOCBFs to minimally modify a nominal robust controller $k(x)$, derived in the next section, so as to ensure the resulting control input u^* is safe. We can also express the safe input as $u^*(x) = k(x) + \delta_u(x)$, regarding δ_u as the extra action needed to satisfy the constraints. Such a task can easily be achieved by solving the following quadratic optimization problem.

$$\begin{aligned} u^*(x) = &\underset{u}{\operatorname{argmin}} \; ||u - k(x)||^2 \\ &\text{s.t.} \quad u \in \mathcal{U}'_s(x) \end{aligned} \tag{11}$$

This control redesign approach, which involves an optimization-in-the-loop, is known as safety filter. Note that u is bounded by the set of admissible control actions $\mathcal{U}$, and thus the optimization problem could in some cases reach an infeasibility. An infeasibility is a signal that can be leveraged, as it has a very clear meaning: there is no input action available that can retain the launcher within safe limits. Such a signal is extremely valuable for safety monitoring, autonomous flight termination systems and guidance re-computation algorithms. It facilitates the supervision of safety constraints and supports autonomous decision-making, contributing to improved safety for populations in proximity to launch sites.

Remark 2.4: *It is possible to avoid dealing with an optimization-in-the-loop if the control input is left unconstrained, i.e.* $u \in \{v \in \mathrm{R}^m : \varphi_r'(x, v) \geq 0\}$. *Although challenging, safe control can be achieved in many cases without saturation by finding appropriate functions* $\alpha_1, ..., \alpha_r$. *Then, Sontag's universal-type formulas apply [19], which provide closed-form solutions to* (11).

3 The Safe TVC Problem

In this section, we describe the two main building blocks of the proposed control loop interconnection for the launcher, depicted in Fig. 1, which consists of a robust H_∞ controller denoted $K(s)$ in series with a robust HOCBF-based safety filter. The aim of this design is to derive TVC inputs with robust stability, performance and safety characteristics, all of which will be detailed in the following sections. To that end, linearized and validated models of the remaining components are presented in Appendix 6.1. Furthermore, we discuss the implications of an existing tradeoff between stability and safety, which highly influences the design of the safety filter. This tradeoff is crucial in safety-critical systems with unstable dynamics, such as launchers and missiles.

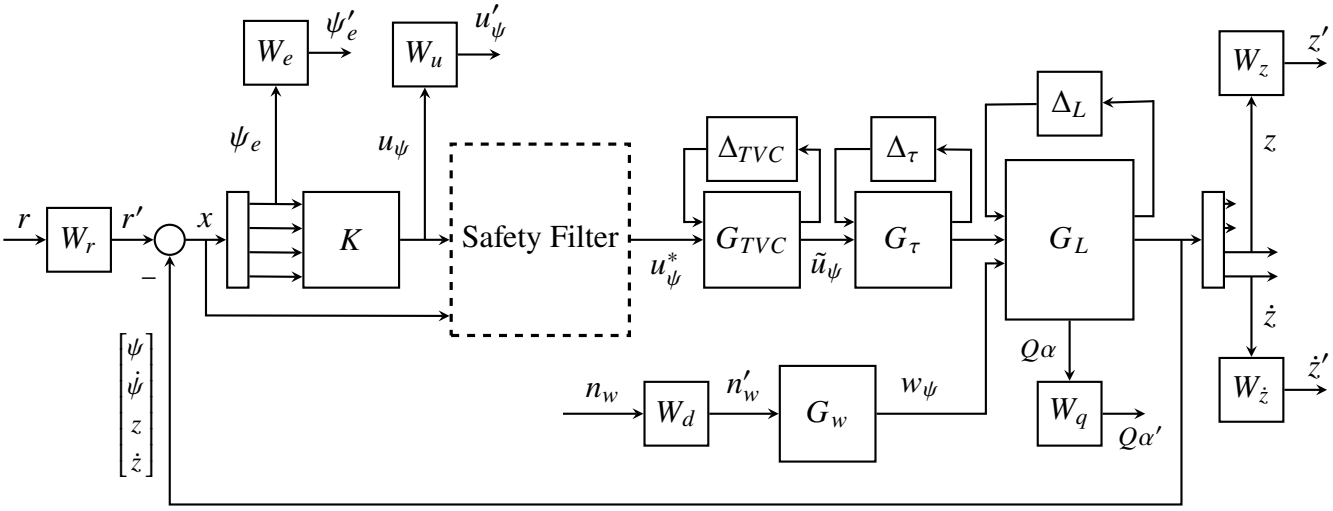


Fig. 1 Proposed control loop interconnection with Safety Filter.

3.1 Nominal Controller

The proposed nominal controller $K(s)$ is based on a structured PD controller tuned using robust H_∞ methods, thereby providing control inputs with robust stability and performance guarantees. Its design is described here for completeness, though in short, since it is not the primary goal of this work. The detailed approach is thoroughly outlined in [20]. Note that this control strategy is selected for its representation of the current state of practice [1], but other alternatives also apply; for instance, those focused on performance (e.g. LQR controllers). Whichever is the chosen strategy, the synthesis is conducted while disregarding the presence of the safety filter block in the control loop interconnection, thus considering that the output of $K(s)$ goes straight to $G_{TVC}(s)$. The safety filter is inserted a posteriori.

Moreover, we focus the design of the controller on a single channel, in this case the yaw channel, since the axial symmetry of the launch vehicle permits to decouple the dynamics between the different attitude channels (yaw, pitch and roll), provided the roll rate is low. For this reason, the interconnection in Fig. 1 and the linear time-invariant (LTI) models in Appendix 6.1 are based on the dynamics around yaw. The controller $K(s)$ then computes robust torque commands in the yaw plane u_ψ. Note that the same approach can be applied to derive a controller for computing robust torque commands in the pitch plane u_θ, which is implicitly accounted for. A robust controller for the roll channel is assumed, yet no further discussion will be provided since this channel plays no role in the proposed safety filter.

Henceforth, the control law in the yaw channel is:

$$u_\psi = k(x,t) = K_p(t) \begin{bmatrix} \psi \\ z \end{bmatrix} + K_d(t) \begin{bmatrix} \dot{\psi} \\ \dot{z} \end{bmatrix} \tag{12}$$

where $K_p(t) = [K_\psi(t)\ K_z(t)]$ are the proportional gains, and $K_d(t) = [K_{\dot{\psi}}(t)\ K_{\dot{z}}(t)]$ the derivative gains. These gains depend on the ascent time t in order to account for the highly dynamic environment. Thus, their values are determined based on a gain-scheduling H_∞ approach [21], using 15 points spaced at equal intervals along the reference trajectory and interpolating at the end. This allows the controller to meet various performance requirements specified by the weight functions $(W_e(s), W_z(s), W_{\dot{z}}(s), W_q(s))$, some of which change throughout the trajectory. Note that this approach involves a familiar tradeoff, whereby the performance weights may have to be relaxed to maintain stability under the unstructured uncertainties considered in the models. A comprehensive explanation of this fundamental limitation can be found in Chapter 6 of [22].

3.2 Safe Control Redesign

The nominal controller $K(s)$, designed for the interconnected system in Fig. 1, computes TVC torque commands (e.g. $u_\psi = k(x,t)$ for the yaw channel) that are not necessarily safe in the CBF sense described in Section 2, i.e. they do not satisfy an inequality like (10). The original task of the safety filter is to redesign the input command $u \mapsto u^*$ so that u^* maintains the stability and performance properties of u while explicitly preventing the launcher from exiting the corridor. This remap is performed through the optimization problem introduced in (11), under appropriately defined functions $\varphi_r'(x,u)$, which will be derived here. However, forcing u^* to exhibit all of these properties simultaneously at all times is fundamentally impossible due to the different tradeoffs present in the design of the proposed control system. We have already made reference to the well-known tradeoff between robust stability and performance of the nominal controller $K(s)$ in the previous section. Actually, a similar tradeoff exists between stability and safety induced by the safety filter. This tradeoff will be detailed in this section, as well as how we addressed it in the design.

Before formally introducing the design considerations that stem from this tradeoff, it is instrumental to ponder how a *naïve* approach to design the safety filter would look like, and its impact on the launcher's behavior. First, note that a set of HOCBFs (in particular, second-order) shall be defined to enforce bounds on the drift states z and y, which are the launcher's states directly connected to the flight envelope constraints. A simple example is via the following half-space constraints, which dynamically bound the upper and lower values of the drift along z so that $z(t) \in [a(t), b(t)]$ at all times (equivalently on y):

$$\begin{aligned} h_{-z}(x) &= z - a \geq 0, \\ h_{+z}(x) &= b - z \geq 0, \end{aligned} \tag{13}$$

Remark 3.1: *Alternatively, one may consider a circular constraint on the drift along both z and y, which in principle reduces the number of constraints from four to one. We discuss this case in the Appendix 6.2.*

Remark 3.2: *The flight envelope can have a complex definition that lacks a closed-form expression, and thus, matching its shape and interior using state constraints such as $(h_{-z}, h_{+z}, h_{-y}, h_{+y})$ might turn overly complicated. Nonetheless, it is generally easier and more efficient to use the state constraints to define a smaller and simpler region within the flight envelope. If the smaller region is rendered forward invariant through CBFs, and therefore safe, then the original flight envelope is also safe provided the launcher starts in this region.*

Then, the robust second-order CBF conditions governing the rate of change of the constraints (13) are derived according to (9). Assuming class-$\mathcal{K}$ functions $\alpha_i(k) = k\alpha_i$ using a slight abuse of notation, with $\alpha_i > 0$ for $i \in \{1, 2\}$:

$$\varphi_{-z}(x, u_\psi) = \frac{-N_\alpha}{mV}\dot{z} + \frac{F}{m}\psi - \frac{N_\alpha \ell_\alpha}{mV}\dot{\psi} + \frac{1}{m\ell_T}u_\psi + \frac{N_\alpha}{mV}w_\infty - \ddot{a} + (\alpha_1 + \alpha_2)(\dot{z} - \dot{a}) + \alpha_1\alpha_2(z - a) \geq 0,$$
$$\varphi_{+z}(x, u_\psi) = \frac{N_\alpha}{mV}\dot{z} - \frac{F}{m}\psi + \frac{N_\alpha \ell_\alpha}{mV}\dot{\psi} - \frac{1}{m\ell_T}u_\psi + \frac{N_\alpha}{mV}w_\infty + \ddot{b} + (\alpha_1 + \alpha_2)(\dot{b} - \dot{z}) + \alpha_1\alpha_2(b - z) \geq 0.$$

The term $\ddot{z}$ has been replaced by the respective dynamics equation from (18) to show the explicit dependency on the control input u_ψ. In practice, the model parameters can be interpolated using the different LTIs used to synthesize $K(s)$ along the trajectory. Moreover, notice that the wind disturbance term has been replaced by the extra buffer term $\frac{N_\alpha}{mV}w_\infty$, according to Section 2.2, to make the conditions robust. Equivalent conditions shall be derived for y, which yield φ_{-y} and φ_{+y} assuming the bounds are identical to z.

To demonstrate this concept moving forward, we consider a simpler case where the bounds form a fixed square centered at $(z, y) = (0, 0)$ that does not change with time, i.e. $a(t) = -b(t) = \gamma_1$, where γ_1 is a positive constant. In this case, $\dot{a} = \dot{b} = 0$, for which the conditions further simplify.

$$\begin{aligned}
\varphi_{-z}(x, u_\psi) &= \left(\alpha_1 + \alpha_2 - \frac{N_\alpha}{mV}\right)\dot{z} + \frac{F}{m}\psi - \frac{N_\alpha \ell_\alpha}{mV}\dot{\psi} + \frac{1}{m\ell_T}u_\psi + \frac{N_\alpha}{mV}w_\infty + \alpha_1\alpha_2(z - \gamma_1) \geq 0, \\
\varphi_{+z}(x, u_\psi) &= \left(\frac{N_\alpha}{mV} - \alpha_1 - \alpha_2\right)\dot{z} - \frac{F}{m}\psi + \frac{N_\alpha \ell_\alpha}{mV}\dot{\psi} - \frac{1}{m\ell_T}u_\psi + \frac{N_\alpha}{mV}w_\infty - \alpha_1\alpha_2(z - \gamma_1) \geq 0.
\end{aligned} \tag{14}$$

Altogether, the two inequalities derived above for z, and the respective versions for y, are the conditions that the nominal input u must satisfy at each instant to be considered safe (in the CBF sense) with regards to the flight corridor. If satisfied, the safety filter maps u to itself, and thus $u^* = k(x, t)$ as it follows from (11). However, let us bring the attention to what occurs when u does not satisfy all the inequalities. In this case, the solution to the optimization problem is a modified version of the nominal control u obtained by projecting it orthogonally to the boundary of the feasibility space defined by all four inequalities. Two insights originate from this:

1) The modification of u to obtain the safe version u^* is *minimal* (with respect to the 2-norm).
2) The modification may lead to a degradation of stability associated with u^* compared to u.

It is natural for this initial setup to inevitably lead to severe degradations in stability since the map $u \mapsto u^*$, executed by the safety filter, has no encoded notion of stability; it only enforces and prioritizes safety constraints. For many systems with marginal or small degrees of instability, this fact can be neglected without major consequences. However, in our particular case concerning a launcher flight corridor, this becomes a critical factor with further implications. On one hand, stability in missiles and launch vehicles is inherently related to the attitude states, ψ and θ, i.e. yaw and pitch. Examining the dynamics equations in (18), it is easy to see that their rate of change exhibit a large positive eigen value. This means that perturbations cause a rapid exponential divergence in ψ and θ, making the recovery of these states harder without fast and precise control. On the other hand, note that any modification to the nominal control input can be regarded as an extra perturbing signal for the launcher. For instance, the safety filter may compute a new control u^* that differs from the nominal u, and thus $\delta_u = u^* - u \neq 0$. This extra signal, δ_u, is meant to guarantee the constraints on the drift, h_{-z}, ..., h_{+y}, but it will also unavoidably affect the attitude states, due to the couplings in the dynamics. Because the current safety filter setup does not incorporate any notion of stability, the severity by which δ_u affects ψ and θ is unpredictable and, therefore, dangerous; specially, considering the instability nature of the attitude states previously mentioned.

3.2.1 *Handling stability and the Stability-Safety tradeoff*

We propose a simple extension to the safety filter that handles the stability considerations described above in a heuristic and straightforward manner. In general terms, this extension aims at bounding the attitude states, ψ and θ, close enough to their respective references to prevent them from diverging largely and quickly. To that end, we include second-order CBFs for the attitude states following the same procedure outlined before for the lateral drift states. For the sake of brevity, here we already consider constant and equal bounds, i.e. $\psi(t) \in [-\gamma_2, \gamma_2]$, where γ_2 is a positive constant (equivalently on θ).

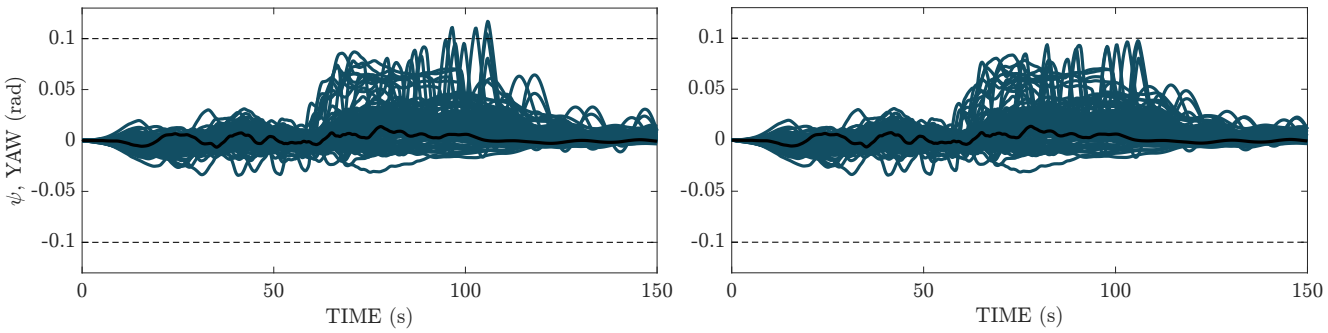
$$\begin{aligned} h_{-\psi}(x) &= \psi - (-\gamma_2) \geq 0, \\ h_{+\psi}(x) &= \gamma_2 - \psi \geq 0, \end{aligned} \tag{15}$$

For this collection of attitude constraints, and assuming class-$\mathcal{K}$ functions $\beta_i(k) = k\beta_i$ with $\beta_i > 0$ for $i \in \{1, 2\}$, the robust second-order CBF conditions are found to be:

$$\begin{aligned} \varphi_{-\psi}(x, u_\psi) &= \left(\beta_1 + \beta_2 - \frac{N_\alpha \ell_\alpha^2}{I_y V}\right)\dot{\psi} - \frac{N_\alpha}{I_y V}\dot{z} + \frac{N_\alpha \ell_\alpha}{I_y V}\psi + \frac{1}{I_y}u_\psi + \frac{N_\alpha \ell_\alpha}{I_y V}w_\infty + \beta_1\beta_2(\psi - \gamma_2) \geq 0, \\ \varphi_{+\psi}(x, u_\psi) &= \left(\frac{N_\alpha \ell_\alpha^2}{I_y V} - \beta_1 - \beta_2\right)\dot{\psi} + \frac{N_\alpha}{I_y V}\dot{z} - \frac{N_\alpha \ell_\alpha}{I_y V}\psi - \frac{1}{I_y}u_\psi + \frac{N_\alpha \ell_\alpha}{I_y V}w_\infty - \beta_1\beta_2(\psi - \gamma_2) \geq 0, \end{aligned} \tag{16}$$

And as before, the two inequalities derived above for ψ, together with the analogous versions for θ, i.e. $\varphi_{-\theta}$ and $\varphi_{+\theta}$, are the conditions that the control input u_ψ and u_θ must enforce to guarantee $h_{-\psi}, \ldots, h_{+\varphi}$.

Let us bring the attention briefly to the new wind buffer term, $\frac{N_\alpha \ell_\alpha}{I_y V}w_\infty$, which was included to robustify the CBF conditions. Provided the attitude states are more sensitive to disturbances, and these can easily lead to constraint violations, the benefits of this buffer term are more notable here than in the drift case. To demonstrate this, Fig. 2 compares the evolution of the yaw angle in two Monte Carlo campaigns, differing only on the CBF conditions for the attitude used in safety filter: Fig. 2a shows a standard safety filter (CBF conditions without a buffer term), and Fig. 2b shows a robust safety filter (CBF conditions with a buffer term). The differing CBF conditions are based on a CBF like (15) with $\gamma_2 = 0.1$ rad. By examining both figures, it is evident that the additional robust term effectively prevents the attitude states from violating the constraints. Indeed, this holds true under the assumption that the models and the wind upper bound w_∞ are representative of the actual scenario.



(a) Standard safety filter **(b) Robust safety filter**

Fig. 2 Comparison of yaw response based on inclusion of the wind buffer term in the CBF conditions.

Finally, note that the CBFs in (15) are just meant to prevent the launcher from becoming unstable, whereas the CBFs in (13) are meant to prevent the launcher from becoming unsafe. The effectiveness of each group of CBFs depends on multiple factors, but most significantly on the design parameters $p = (\alpha_1, \alpha_2, \beta_1, \beta_2)$ and the bounds $\gamma = (\gamma_1, \gamma_2)$. In general, it is complicated and nonintuitive to *tune*

CBFs through the class-$\mathcal{K}$ functions, even after simplifying them as performed here. Usually, large values in p lead to an aggressive safety filter, allowing the launcher to get closer to the boundary of the constraints but also exhibiting an undesired wobbling behavior. Smaller values in p, on the contrary, lead to a more gentle safety filter that may reduce the wobbling behavior, but it also yields a more intrusive safety filter as it *senses* the boundary from further away.

On the other side, the boundary values in γ have different interpretations. Clearly, the value of γ_1 determines the size of the *safety* corridor, which, according to the drift constraints (13), define a transversal square of side length $2\gamma_1$ around the reference trajectory. It may be desired to maximize γ_1 such that the region defined by these constraints gets as close to the actual flight corridor as possible, but without exceeding its boundaries. The larger the region defined by the safety constraints, the less intrusive the safety filter will be. Similarly, the parameter γ_2 defines the size of the operating region for the attitude states, but it also balances two competing objectives: stability and safety. A larger γ_2 enhances the controller's flexibility by allowing greater deviations from the attitude's reference, enabling the pursuit of secondary objectives such as minimizing structural loads or, as emphasized in this work, limiting lateral drift. However, this comes at the cost of increased risk of instability, as the wider bounds may permit the safety filter to push attitude states to unrecoverable levels or induce excessive structural loads. Conversely, a smaller γ_2 tightens the feasible region, promoting stability by keeping attitude states closer to the reference, but it restricts the controller's ability to address secondary objectives.

The previous discussion provides some insights on the relevancy of these parameters; specially on γ_2, which mainly governs the tradeoff between stability and safety. Designing a procedure to determine *optimal* values for p and γ is out of the scope of this work. Instead, a trial-and-error iterative approach has been used to find appropriate values for each parameter. The final values are reported in Section 4.

Remark 3.3: *Note then, that collecting the conditions* (14) *and* (16) *as the constraints for the safety filter's optimization problem might sometimes lead to an infeasibility, which carries meaningful information: there are no available control actions that prevents the launcher from violating the safe constraints* (13)*. Monitoring whether the infeasibility occurs can be key for other systems, e.g. an Autonomous Flight Termination system or an Adaptive Guidance system.*

3.2.2 Launcher Safety Filter Formulation

In this final section, we formulate the resulting optimization problem that will be solved at each step of the trajectory within the safety filter block. The optimization is based on the formulation introduced in (11), with the constraints replaced by the CBF conditions derived thus far. Since the *safety* corridor defined by (13) is conservative (i.e. much smaller than the actual corridor), it may be preferable to relax one of the two groups of CBF conditions, either those associated to the lateral drift or those associated to the attitude, to avoid dealing with infeasible solutions. Based on the previous discussion, relaxing the attitude conditions would lead to a relaxation in stability, potentially causing a catastrophic loss of the vehicle. Instead, it is proposed to relax those associated to the lateral drift, thereby sacrificing safety capabilities to preserve stability. Recall that γ_2 is the main parameter controlling this tradeoff.

The overall relaxed optimization problem for the yaw channel reads as follows (equivalently for the pitch channel):

$$
\begin{aligned}
\begin{bmatrix} u_\psi^* \\ \delta_\psi^* \end{bmatrix} &= \underset{u_\psi, \delta_\psi}{\operatorname{argmin}} \; ||u_\psi - k(x,t)||^2 + M\delta_\psi \\
\text{s.t.} \quad & u_\psi \in \mathcal{U}, \quad \delta_\psi \geq 0 \\
& \varphi_{-z}(x, u_\psi) \geq \delta_\psi, \quad \varphi_{+z}(x, u_\psi) \geq \delta_\psi, \\
& \varphi_{-\psi}(x, u_\psi) \geq 0, \quad \varphi_{+\psi}(x, u_\psi) \geq 0,
\end{aligned}
\tag{17}
$$

where M is a large enough positive scalar penalizing the slack variable δ_ψ. An identical optimization problem to (17) is formulated for the pitch channel, with the corresponding CBF conditions and analogous variables. Given the decoupling assumption, both optimization problems can be formulated separately.

Remark 3.4: *This formulation allows the lateral drift constraints to be violated when both, attitude and drift, are at their limits. An outcome of this relaxation is the slack variable δ_ψ, which can be considered to have acceleration units by inspecting the CBF conditions. If M is large enough, then δ_ψ is nonzero only when the optimization problem with hard constraints is unfeasible. Therefore, similar conclusions to those in Remark 3.3 can be drawn, making δ_ψ a key component for safety operations. For instance, it can be used to monitor the extent of violation of the corridor constraints in extreme cases, informing operators about the launcher's excess of lateral acceleration during the violation.*

4 Simulation and Results

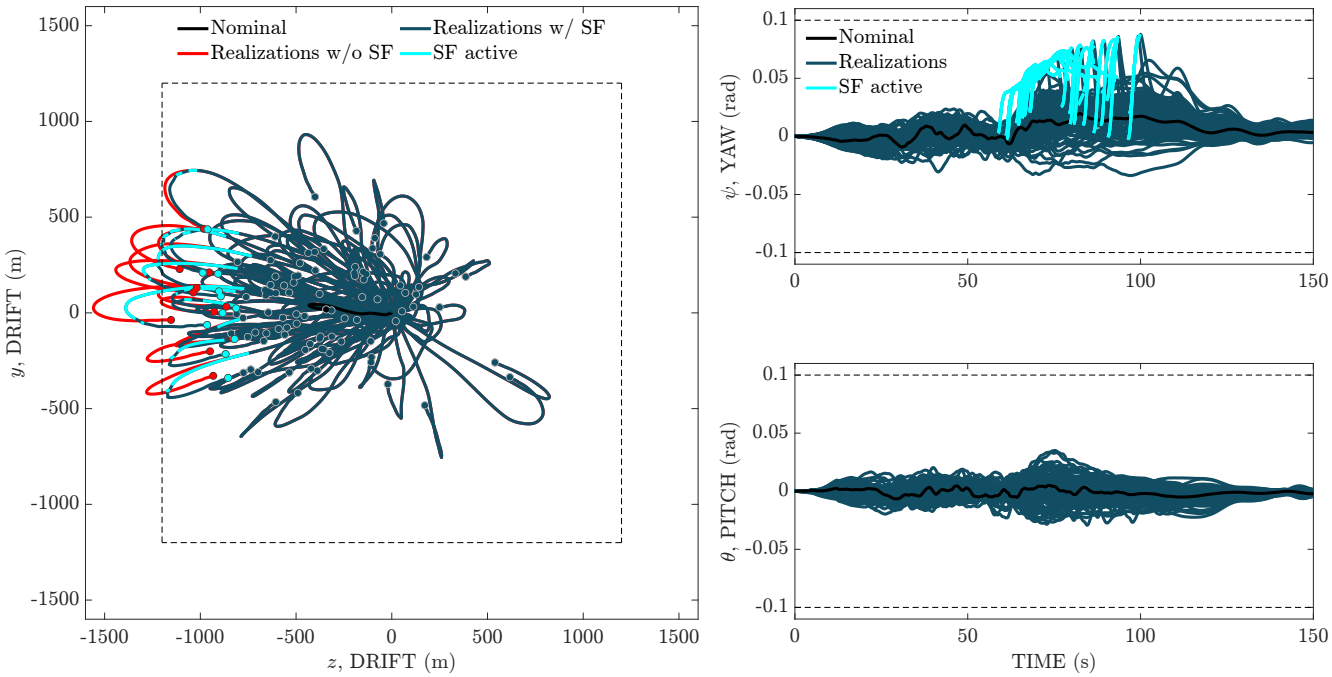
The performance of the second-order robust safety filter was demonstrated in a Monte Carlo campaign conducted in a high-fidelity simulator, the ESA-i4GNC framework [23], which includes accurate aerodynamic, propulsion, wind and turbulence models. A total of 100 simulations were performed based on the ascent flight scenario of a two-stage launcher, up until main engine cut-off (150 seconds), allowing dispersions in the TVC parameters, the launcher's mass, center of gravity and moments of inertia, and the aerodynamic coefficients. The launcher model within the simulator is populated with realistic parameters provided by Orbex, and considers a reference trajectory with Sutherland spaceport as launch site. The navigation is assumed ideal, and the study of its effects on the safety filter and the overall control loop is left out of the scope of this work. The results presented here are extended in a Software-in-the-Loop (SIL) validation environment in [24], bringing the proposed approach to an estimated TRL of 3-4.

The values chosen for the safety filter's design parameters are summarized in Table 1.

Table 1 Safety Filter design parameters

Parameter	Value	Description
α_1	0.3	Tune the aggressiveness of the Safety Filter
α_2	0.1	
β_1	7	
β_2	2	
γ_1	1200 (m)	Maximum lateral drift – Defines the side length of the safety corridor's cross section
γ_2	0.1 (rad)	Maximum attitude error – Controls the tradeoff between stability and safety
w_∞	50 (m/s)	Maximum expected windspeed – Defines the buffer sizes on the robust CBF conditions
M	10^6	Penalization on the slack variable

The results presented in Fig. 3, show the time evolution of the relevant states in the yaw and pitch channels obtained in the simulations. In Fig. 3a, the evolution of the lateral drift associated to two different Monte Carlo campaigns are superposed; one where the safety filter is disabled (displayed in red–) and another one where it is enabled (displayed in teal– and cyan–). Two colors are used in the latter to differentiate when the safety filter is active (i.e. modifying the nominal controller) from when it is not. These results demonstrate a series of features, listed below:



(a) Lateral drift with and without safety filter. **(b) Attitude error with safety filter**

Fig. 3 Evolution of the lateral drift and attitude error states during ascent, emphasizing the time instances where the safety filter modifies the nominal controller.

- The safety filter reduced the lateral drift of extreme cases, resulting in smaller orbit injection errors.
- The safety filter reduced the number of cases exceeding the safety constraints from 9 to only 3.
- The safety filter became active only in those cases where inaction would have led to a violation (or close violation) of the safety constraints.

The latter point demonstrates the unobtrusive behavior of the safety filter, allowing the nominal controller to exhibit its robustness and performance properties most of the time, as long as safety is not compromised. Fig. 3b presents the evolution of the attitude error in the Monte Carlo campaign with the safety filter enabled, showcasing two relevant aspects:

- The safety filter increases the attitude error in order to reduce the drift, as expected. However, while it allows the lateral drift to cross the drift constraints (soft constraint), it guarantees the satisfaction of the attitude constraints (hard constraint).
- The safe control inputs generated by the safety filter in one channel (yaw) do not affect or pollute other channels (pitch).

The solutions to the optimization-in-the-loop computed within the launcher's safety filter, which yield the results shown above, are presented in Fig. 4. As before, we display these results in two different colors (green– and lime–) to denote if the safety filter is active or not. We show the final TVC deflection angle $\hat{u}_\psi^*$ executed by one of the launcher's engines, based on the torque input commands provided by the safety filter u_ψ^*, and the slack variable δ_ψ^*. We only show the deflections of one out of 6 engines, since they all share similar deflection signals. Observe that while the safety filter occasionally commands fast TVC deflections, the actuation manages to execute them without saturating or unstabilizing the system. Furthermore, note that the slack variable becomes nonzero in several realizations, revealing that the CBF conditions are violated in multiple simulations. This seems counterintuitive, since we previously saw that only 3 cases ended up exceeding the lateral drift (safety) constraints. To clarify this, it is important to emphasize the distinction between the safety constraints (or CBFs) and the CBF conditions. While the satisfaction of the CBF conditions guarantees the satisfaction of the safety constraint, the opposite is not necessarily true.

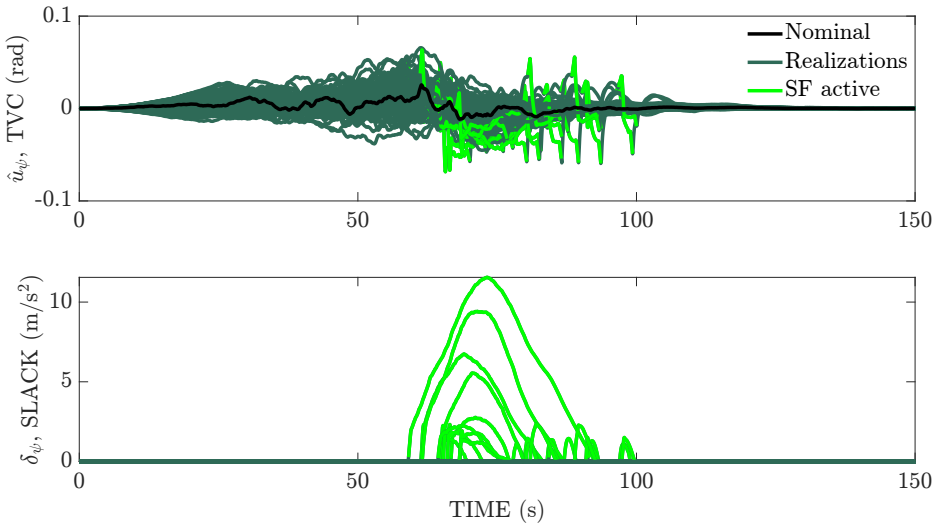


Fig. 4 Solutions yield by the safety filter

The possibility of using the slack variable as a safety monitoring signal was previously discussed in Remark 3.4. From the inspection of Fig. 4, it is observed that the three (or four) most critical realizations are readily identified based solely on this variable, all of which resulted in (or were very close to) constraint violations. Based on this observation, a very simple rule of thumb could be defined as $\delta_\psi^* > 5$ m/s^2, effectively separating the most dangerous cases that the safety filter may not be able to accommodate. Indeed, more complex and sophisticated strategies could be developed instead, so as to process δ_ψ^* in order to take more informative decisions. Finally, note that considering δ_ψ to have units of m/s^2, implicitly assumes that the tuning parameters α_1, α_2 have units of 1/s. Such considerations may offer some intuition on how to properly tune the safety filter.

As a final step, we compare the increase in structural loads caused by the safety filter in Fig. 5. This increase is expected, due to the attitude corrections performed to prevent the growth of the lateral drift, and the fact that the safety filter was not explicitly designed with load-relief objectives. In fact, it is foreseeable that the parameter γ_2 influences not only the tradeoff between stability and safety, but also between safety and load handling. Nonetheless, observe that the additional loads induced by the safety filter occur after the point of maximum dynamic pressure (at approx. 60s), and thus they never exceed the maximum load requirement.

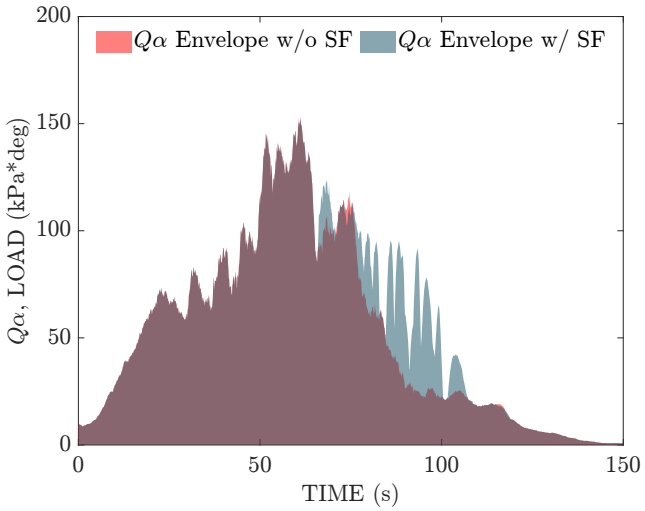


Fig. 5 $Q\alpha$ Load factor

5 Conclusions

This paper proposes a TVC control loop based on a robust H-infinity controller and a safety filter to provide robust stability, performance, and safety capabilities to launch vehicles. The resulting closed-loop behavior demonstrates two main features: 1) it enables the autonomous handling of critical flight-envelope constraints in an unobtrusive manner, and 2) it achieves this without altering the overall GNC architecture.

In particular, we have presented the development of a CBF-based safety filter in detail and discussed the relevance of the tradeoff introduced by the safety filter between stability and safety. We have argued that, in this class of unstable critical systems, stability must take precedence over strict safety enforcement. This consideration has shaped the design of the safety filter, leading to a formulation that allows temporary violations of the safety constraint in order to preserve stability. As a consequence of this tradeoff, the formal stability and safety guarantees associated with the robust controller and the safety filter cannot be simultaneously maintained while the safety filter is actively resolving this conflict. Methods to recover formal stability and safety certificates for the overall control loop, in order to support verification and validation of the proposed architecture, are currently under investigation. Finally, it has been shown that the violations of the safety constraints yield a slack variable containing meaningful information, whose value can be used as a safety monitoring signal.

These results represent a promising step toward the integration of safety-critical control architectures in launch vehicle GNC systems, bringing the proposed approach to an estimated TRL of 3-4 and motivating further work toward autonomous safety verification and enforcement within flight control systems.

6 Appendix

6.1 Linearized Models

For completeness, we briefly describe the dynamic models of the launcher, rocket actuation, and the wind perturbation, as well as the diverse assumptions proposed to facilitate the synthesis of the nominal controller. The validation of the LTI models presented here was made in [25].

6.1.1 Launcher Model

The launcher model equations used for the control design are based on the short-period, rigid-body dynamics [26]. These equations assume small deviations from a given reference trajectory, which enable a state space representation of the differential equations. Furthermore, the decoupling assumption allows us to formulate the equations of motion independently on each attitude channel, as previously mentioned. The following linear system describes thus the dynamics only on the yaw plane, and the nominal and safe control design are derived with regards to this decoupled, linearized model. Under this assumption, the control design with respect to the pitch channel follows exactly the same principles.

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{N_\alpha \ell_\alpha}{I_y} & \frac{-N_\alpha \ell_\alpha^2}{I_y V} & 0 & \frac{-N_\alpha \ell_\alpha}{I_y V} \\ 0 & 0 & 0 & 1 \\ \frac{F}{m} & \frac{N_\alpha \ell_\alpha}{mV} & 0 & \frac{-N_\alpha}{mV} \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_y} & \frac{N_\alpha \ell_\alpha}{V I_y} \\ 0 & 0 \\ \frac{1}{m \ell_T} & \frac{N_\alpha}{mV} \end{bmatrix} \begin{bmatrix} u_\psi \\ w_\psi \end{bmatrix} \tag{18}$$

Here, z is the lateral displacement or drift, and ψ is the yaw angle error. The control input u_ψ is the torque generated by the thrust force of the actuated gimbal rocket nozzle and the input w_ψ is the perturbing wind speed, both in the yaw plane. The physical parameters are the total mass m, the inertia in the y-axis I_y, the total airspeed V, the distance between the center of gravity and center of pressure ℓ_α, the distance

between the center of gravity and the thrust point ℓ_T, and the resultant force $F = D - N_\alpha - T$, where D is the drag, T is the thrust and N_α is the aerodynamic force gradient with respect to the incident angle α.

The output equation, which will allow us to express requirements regarding reference tracking error and structural loads, is then

$$\begin{bmatrix} \psi \\ \dot{\psi} \\ z \\ \dot{z} \\ Q\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ Q & 0 & 0 & \frac{Q}{V} \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{Q}{V} \end{bmatrix} \begin{bmatrix} u_\psi \\ w_\psi \end{bmatrix} \tag{19}$$

Due to the variation of many of these parameters in the model, e.g. in mass, inertia, windspeed and aerodynamic components, these LTI equations are only valid for a short period of time. This caveat will be addressed in the control design using a classical gain-scheduling approach. Since we synthesise a robust H_∞ controller, it is common to express the models in the frequency domain. Therefore, we denote the matrix transfer function of the state space model described above, i.e. (18) and (19), as $G_L(s)$.

Some of the physical parameters are also considered uncertain, including the moment of inertia I_y, the distances ℓ_α and ℓ_T, the airspeed V, the forces D and F_α, and the dynamic pressure Q. These parametric uncertainties will be introduced through an upper linear fractional transformation (LFT), with unstructured uncertainty denoted Δ_L.

6.1.2 TVC Actuation Model

The thrust vector control (TVC) actuation dynamics are applied to deflection angles, $\hat{u}$, rather than torques. Since both the controller output and the system input are assumed to be torques, the TVC dynamics should also capture the conversion from control torque reference u, to TVC deflection angle $\hat{u}$ and back to torque $\tilde{u}$. This is modeled as a second-order system with the following transfer function.

$$G_{TVC}(s) = \frac{K_a \omega_a^2}{s^2 + 2\xi_a \omega_a s + \omega_a^2}, \tag{20}$$

where K_a is the low-frequency gain, ω_a the natural frequency and ξ_a the dampeing ratio. All these parameters are assumed uncertain, and represented by an LFT interconnection with uncertainty Δ_{TVC}.

6.1.3 Delay Model

The compounded delay of the overall system is approximated by a first-order Padé model, which corresponds to the following transfer function.

$$G_\tau(s) = \frac{1 - 0.5\tau s}{1 + 0.5\tau s} \tag{21}$$

where the delay parameter τ is also assumed uncertain and represented by an LFT with uncertainty Δ_τ.

6.1.4 Wind Model

The launcher's stability and structural loads during the atmospheric flight are primarily affected by wind perturbations. To capture the impact of the wind on the system we use a model often found in literature that can be integrated in the control synthesis. This corresponds to the following second-order

Dryden filter [27]:

$$G_w(s) = \sigma_h \sqrt{\frac{V_h}{L_h}} \frac{\sqrt{\frac{3}{\pi}} s + \frac{1}{\sqrt{\pi}} \frac{V_h}{L_h}}{s^2 + 2\frac{V_h}{L_h} s + \left(\frac{V_h}{L_h}\right)^2} \tag{22}$$

where $V_h = V - v_{wp}(h)$ with v_{wp} being the mean of the low-frequency vertical wind profiles, $\sigma_h = \sigma(h)$ is the standard deviation and $L_h = L(h)$ is the turbulence length scale, for a given altitude value h. This approximation captures the turbulent component of the wind for control synthesis at the different altitudes. The low-frequency component of the wind is handled by the drift z and drift rate $\dot{z}$ feedback.

6.2 Alternative Lateral Drift Constraint

An alternative constraint to bound the lateral drift can be considered as follows.

$$h(x) = \frac{1}{2}(\gamma^2 - y^2 - z^2) \geq 0, \tag{23}$$

where γ is a known signal or scalar function of time, bounding the drift radially at each instant. At first glance, this seems appropriate as it reduces the four constraints we introduced to only one. Assuming $\alpha_i(k) = k\alpha_i$ with $\alpha_i \in \mathbb{R}$ for $i \in \{1, 2\}$, the preliminary second-order CBF condition can be found to be:

$$\varphi_2(x, u) = \dot{\gamma}^2 - \dot{y}^2 - \dot{z}^2 + \gamma\ddot{\gamma} - y\ddot{y} - z\ddot{z} - (\alpha_1 + \alpha_2)(\dot{\gamma}\gamma - \dot{y}y - \dot{z}z) + \frac{\alpha_1\alpha_2}{2}(\gamma^2 - y^2 - z^2) \geq 0, \tag{24}$$

Note that the control u is implicit in the terms $\ddot{z}$ and $\ddot{y}$, however, these terms disappear from (24) if $(z, y) = (0, 0)$. When this occurs, the remaining states might fall into configurations where the CBF inequality may not longer hold, thereby preventing the constraint (23) from actually being a CBF. Nonetheless, in practice one may be able to disregard this fact, since the condition $z, y = 0$ occurs at the center of the constraint where the launcher is safest. Moreover, it could be argued that the time spent at those points is negligible. At any rate, this setup would still carry the burden of handling the points where the control is cancelled and the CBF conditions are not satisfied, which would inevitably lead to infeasible solutions. For these reasons, it was decided not to continue with this choice for the lateral drift constraint.

Acknowledgments

The results presented in this paper have been achieved under the SURE project. The SURE project is funded by the European Space Agency through Contract No. 4000142645/23/NL/CRS. The views expressed in this paper can in no way be taken to reflect the official opinion of the European Space Agency.

Declaration of Use of Artificial Intelligence

Artificial intelligence was not used in the work presented.

References

[1] A. Marcos, D. Navarro-Tapia, P. Simplício, and S. Bennani. Robust Control for Launchers: VEGA Study Case. *Journal of The Society of Instrument and Control Engineers*, vol. 59(3):192–202, 2020. doi: 10.11499/sicejl.59.192.

[2] D. Navarro-Tapia, A. Marcos, S. Bennani, and C. Roux. Linear Parameter Varying Control Synthesis for the atmospheric phase VEGA launcher. *IFAC-PapersOnLine*, vol. 51(26):68–73, January 2018. doi: 10.1016/j.ifacol.2018.11.166.

[3] A. Marcos, S. Bennani, C. Roux, and M. Valli. LPV modeling and LFT Uncertainty Identification for Robust Analysis: application to the VEGA Launcher during Atmospheric Phase. *IFAC-PapersOnLine*, vol. 48:115–120, January 2015. doi: 10.1016/j.ifacol.2015.11.123.

[4] P. Simplício, S. Bennani, A. Marcos, C. Roux, and X. Lefort. Structured singular-value analysis of the vega launcher in atmospheric flight. *Journal of Guidance, Control, and Dynamics*, vol. 39:1342–1355, June 2016. doi: 10.2514/1.G000335.

[5] Lu Ping. Introducing computational guidance and control. *Journal of Guidance, Control, and Dynamics*, vol. 40(no. 2):193–193, February 2017. doi: 10.2514/1.G002745.

[6] B. Açıkmeşe, J. M. Carson, and L. Blackmore. Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem. *IEEE Transactions on Control Systems Technology*, 21, no. 6:2104–2113, 2013. doi: 10.1109/TCST.2012.2237346.

[7] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, and M. Pavone. Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently. *IEEE Control Systems Magazine*, 45, no. 5:40–113, 2022. doi: 10.1109/MCS.2022.3187542.

[8] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. *In Proc. 18th European Control Conference (ECC)*, page 3420–3431, 2019.

[9] Ian M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. *Lecture Notes in Computer Science, vol. 4416, Springer, Berlin, Heidelberg*, 2007.

[10] Kim P. Wabersich and Melanie N. Zeilinger. Linear model predictive safety certification for learning-based control. *IEEE Conference on Decision and Control (CDC)*, pages 7130–7135, 2018.

[11] W. Xiao, C. G. Cassandras, and C. Belta. Safe autonomy with control barrier functions: Theory and applications. *Synthesis Lectures on Computer Science. Cham: Springer International Publishing*, 2023. doi: 978-013456567510.1007/978-3-031-27576-0.

[12] J. Breeden and D. Panagou. Robust Control Barrier Functions under high relative degree and input constraints for satellite trajectories. *Automatica*, 155:111109, Sept. 2023. doi: 10.1016/j.automatica.2023.111109.

[13] J. Breeden and D. Panagou. Guaranteed Safe Spacecraft Docking with Control Barrier Functions. *IEEE Control Syst. Lett.*, 6:2000–2005, 2022. doi: 10.1109/LCSYS.2021.3136813.

[14] J. Breeden and D. Panagou. Compositions of Multiple Control Barrier Functions Under Input Constraints. *American Control Conference*, pages 3688–3695, 2023.

[15] W. Xiao and C. Belta. Control barrier functions for systems with high relative degree. *Conference on Decision and Control (CDC)*, pages 3655 – 3662, December 2019.

[16] M. Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, vol. 96:359–367, October 2018. doi: 10.1016/j.automatica.2018.07.004.

[17] A. Constantin. On Nagumo's theorem. *Proceedings of The Japan Academy Series A-mathematical Sciences*, 86, 2010. doi: 10.3792/pjaa.86.41.

[18] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. Robustness of Control Barrier Functions for Safety Critical Control. *IFAC-PapersOnLine*, 48, no. 27:54–61, 2015. doi: 10.1016/j.ifacol.2015.11.152.

[19] E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13, no. 2:117–123, 1989. doi: 10.1016/0167-6911(89)90028-5.

[20] J. P. Belfo, B. Ribeiro, G. Videira, A. Botelho, I. Zagalo, P. Guerreiro, A. Montero Miñán, J. Vasconcelos, P. Rosa, Adolfo D. Silva, P. Simplício, and M. Casasco. Robust wind disturbance observer design for a flexible launch vehicle. *11th Symposium on Robust Control Design (ROCOND)*, Porto, Portugal, 2025.

[21] R.A. Hyde and K. Glover. The Application of Scheduled H-inf Controllers to a VSTOL Aircraft. *IEEE Transactions on Automatic Control, 38*, pages 1021–1039, 1993.

[22] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and optimal control*. Prentice Hall, 1996. ISBN: 978-0134565675.

[23] ESA. Final Report, Artificial Intelligence techniques for GNC design, implementation and verification. Technical report, Contract 4000134108/21/NL/CRS.

[24] J. P. Belfo, P. Guerreiro, B. Ribeiro, A. Botelho, G. Videira, J. Vasconcelos, P. Rosa, Adolfo D. Silva, A. R. Gomez, J. Stoustrup, P. Simplício, and M. Casasco. Development and Validation of Enabling Guidance and Control Technologies for Smart Launchers. *2026 CEAS EuroGNC Conference*, Madrid, Spain, 2026.

[25] J. P. Belfo, N. Somma, A. Montero, P. Rosa, J. Santos, T. Moreira, P. Simplício, A. Rinalducci, and Y. Torres. Robust control design for a sub-orbital launch vehicle with destabilizing sloshing dynamics. *Proceedings of the 2024 CEAS EuroGNC Conference*, Bristol, UK, 2024.

[26] A. L. Greensite. Analysis and Design of Space Vehicle Flight Control Systems. Volume I - Short Period Dynamics. *GD/C-DDE65-055*, Jan. 1967, Accessed: Oct. 2024. [Online]. Available: https://ntrs.nasa.gov/citations/19670020656.

[27] D. L. Johnson. Terrestrial environment (climatic) criteria guidelines for use in aerospace vehicle development. *NASA Technical Memorandum, NASA TM 4511*, August 1993.