GNC 2017 : 10th International ESA Conference on Guidance, Navigation & Control Systems
29 May - 2 June 2017, Salzburg, Austria
European Space Agency

STRUCTURED H-INFINITY CONTROL DESIGN FOR THE VEGA LAUNCH VEHICLE: RECOVERY OF THE LEGACY CONTROL BEHAVIOUR

Diego Navarro-Tapia[(1)], **Andrés Marcos**[(1)], **Samir Bennani**[(2)], **Christophe Roux**[(3)]

[(1)] *University of Bristol, BS8 1TR, United Kingdom (Technology for AeroSpace Control (TASC), www.tasc-group.com; e-mail: diego.navarro-tapia/andres.marcos @bristol.ac.uk)*
[(2)] *ESA-ESTEC, Noordwijk, 2201AZ, The Netherlands, (e-mail: samir.bennani@esa.int)*
[(3)] *ELV S.p.A., Colleferro, 00034, Italy, (e-mail: christophe.roux@elv.it)*

ABSTRACT

This article addresses the design of the atmospheric ascent-flight control system of a launch vehicle using the structured $\mathcal{H}_\infty$ synthesis technique. The main goal is to provide a methodological robust control design framework and transfer it to industry. This article presents the first step towards this transfer, which consists of: first, providing guidelines to tune the weighting functions based directly on physical effects of the launch vehicle and system requirements; and second, to demonstrate that a baseline, classically-design controller can be recovered with this design framework. The latter is very important for an industrial transfer as it establishes a common point of reference. The system is used based on the actual VEGA vv05 mission data, and classical VEGA TVC structure. These data and structure are then used together with $\mathcal{H}_\infty$ metrics to guide the structured $\mathcal{H}_\infty$ design to recover the same controllers as the baseline along different design points. Finally, the fixed-order $\mathcal{H}_\infty$ controllers are gain-scheduled and validated using a non-linear high-fidelity simulator, which shows the perfect performance and robust recovery.

1 INTRODUCTION

The design of the ascent-flight control system of a launch vehicle for the atmospheric phase is a challenging task. Along this first phase of the mission, the launch vehicle is considerably affected by several undesired effects such as high aerodynamic loads, wind disturbances and rapid system dynamic changes. Further complications are introduced by the elastic behaviour of the launch vehicle, which may cause instability. In the face of all these adverse effects, the control system must satisfy very demanding and tight performance requirements and still be robust against a large range of substantial parameter dispersion. This robustness is generally verified in terms of classical stability margins and validated by exhaustive Monte-Carlo time-domain simulations using a high-fidelity nonlinear simulation model.

As demonstrated by the current state-of-practice, there is a rich heritage and experience in applying classical control solutions to the launcher problem. In order to deal with the wide dynamic variation along the atmospheric phase the classical approach uses the so-called Gain Scheduling (GS) scheme [1]. It consists of linearizing the vehicle around several representative points along the flight trajectory and designing a frozen time controller at each point. These individual controllers are then interpolated based on a parameter (e.g. time or non-gravitational-speed) resulting in a scheduled controller. This is the design approach used by the small European VEGA launcher, which uses a classical (proportional-derivative plus bending filters) controller for the Thrust Vector Control (TVC) system [2].

This strategy has been proven successful for the nine flights VEGA has performed so far but several practical limitations are recognized. First, each linear point design is a complex process which has to account for different concurrent requirements. Using a classical control design framework, the design of multiple-input multiple-output (MIMO) systems becomes very tedious, since every channel/requirement is iteratively addressed in a single-loop fashion. This results in an expensive (in terms of both cost and time) design process. In addition, the tuning process is partially automated, with limited or no connection to other points or across missions. Second, the classical design is performed for nominal conditions, therefore, model uncertainties are not considered in the design process. As a consequence, the performance and robustness verification and validation process relies in an extensive analysis coverage after design (again costly and time intensive).

On the other hand, robust control design techniques generally allow considering uncertainties in the design and are oriented to multivariable control problems [3, 4]. Those are powerful features that can facilitate and improve the design task. Moreover, the robust control theory also permits to evaluate the robustness of the design using analysis tools such as the structured singular value μ. This analytical approach has already been applied to the VEGA launcher in [5, 6], providing relevant insights on the performance degradation due to model uncertainties. One of the major potentials of the μ analysis is that is performed under the same design framework, which provides a direct connection to design.

In order to address the aforementioned state-of-practice limitations, advanced robust control synthesis techniques are being considered within the frame of an ESA Networking Partnering Initiative (NPI No. 4000114460) participated by ESA-ESTEC, ELV and the University of Bristol with the aim to provide a methodological framework for atmospheric launcher control design and transfer it to industry. In this paper, the recently developed structured $\mathcal{H}_\infty$ approach [7] is applied to the VEGA launch vehicle for the design of the atmospheric ascent-flight control. This new approach, which is based on $\mathcal{H}_\infty$ theory, allows defining a specific order and structure for the controller. This technique has shown great promise and has led to intense study by the community, even resulting already in relevant Space flown missions, such as ESA Rosetta [8], CNES Microscope [9], and piloted flight tests [10]. In addition, this technique has also been successfully applied to launch vehicle control design in several investigations [11, 12, 13]. Nonetheless, these applications are being performed by $\mathcal{H}_\infty$ experts, since unlike classical control, $\mathcal{H}_\infty$ requirements are expressed in terms of weighting functions in the frequency domain. The authors feel that a detailed understanding of the $\mathcal{H}_\infty$ metrics, based on the so-called sensitivity functions, reconciled with physical system effects, is required for adequate transfer to industrial control engineers with a more classical control background.

In this paper, as a first step towards a more methodological robust design framework, the structured $\mathcal{H}_\infty$ synthesis is guided to recover the baseline VEGA vv05 mission controller along the different linear design points. This article extends the work done in [14], but instead of a SISO model, in this case a MIMO control problem is considered. The aim is to show how to define the weighting functions in terms of system requirements such as lateral control, load and/or actuation effort.

The layout of this paper is as follows: first, the VEGA launch vehicle and the analytical model for the atmospheric phase are described in Section 2. Secondly, the structured $\mathcal{H}_\infty$ control design is formulated for the VEGA launcher in Section 3. The design process and the weighting function selection to recover the baseline controller are exemplified for a particular flight instant. Then the different structured $\mathcal{H}_\infty$ linear designs are scheduled, implemented and validated in a non linear high-fidelity simulator. Finally, Section 4 ends with the conclusions.

2 VEGA LAUNCHER

2.1 VEGA launcher and mission

VEGA launcher is the new European Small Launch Vehicle developed under the responsibility of the European Space Agency (ESA) and European Launch Vehicle (ELV) as prime contractor. The launcher has successfully performed nine missions since February 2013.
VEGA follows a four-stage approach formed by three solid propellant motors (P80, Zefiro 23 and Zefiro 9) providing thrust for the 1st, 2nd and 3rd stages; and, a bi-propellant liquid engine (LPS) on the 4th stage. All stages are controlled using a TVC. There is also a Roll and Attitude Control System (RACS) performing 3-axes control during the ballistic phase and roll rate control during the propelled phases.

2.2 VEGA model

The motion of the vehicle is described by the standard six-degree-of-freedom equations of motion, which account for the translational and rotational dynamics of the launch vehicle. The derivation of the equations of motion of a generic launch vehicle can be found in [15]. In this article, the dynamic model is built following reference [16], where the equations of motion are expressed in a state-space model which is a suitable representation for analysis and design. In addition, this formulation can also be used to further derive Linear Fractional Transformation (LFT) and Linear Parameter Varying (LPV) models. This extension to LFT/LPV is not covered in this article.
Due to axial symmetry of the vehicle about the roll axis, the design and analysis can be performed in a single plane, either the pitch or the yaw axis. This strategy is valid considering the roll rate negligible so the pitch and yaw axes can be assumed uncoupled. In this work, the VEGA launch vehicle will be examined in the yaw plane.
The translational and rotational equations are expressed as the sum of forces and moments from rigid-body and nozzle motion dynamics, also known as the tail-wags-dog effect (see equations 1 and 2). In addition, other contributions such as rigid damping and wind disturbances are included. Note that for the ease of simplicity in the design process, only the rigid-body motion of the launch vehicle is considered. The design including the flexible-body dynamics will be addressed in future works.

$$m\ddot{z} = \Sigma F = F_r + F_n \tag{1}$$

$$J_{yy}\ddot{\psi} = \Sigma M = M_r + M_n \tag{2}$$

where m is the vehicle mass, J_{yy} is the moment of inertia, $\ddot{z}$ is the linear drift acceleration and $\ddot{\psi}$ the yaw attitude acceleration.
The rigid-body model in Figure 1 describes the vehicle motion due to thrust and aerodynamics. Using small-angle approximations, the rigid-body motion forces F_r and moments M_r are presented in equations 3 and 4 respectively.

$$F_r = -(T - D)\psi + N(\psi + \frac{\dot{z}}{V} + \frac{v_w}{V} - \frac{\dot{\psi}}{V}l_{CP}) - T_c\beta_\psi \tag{3}$$

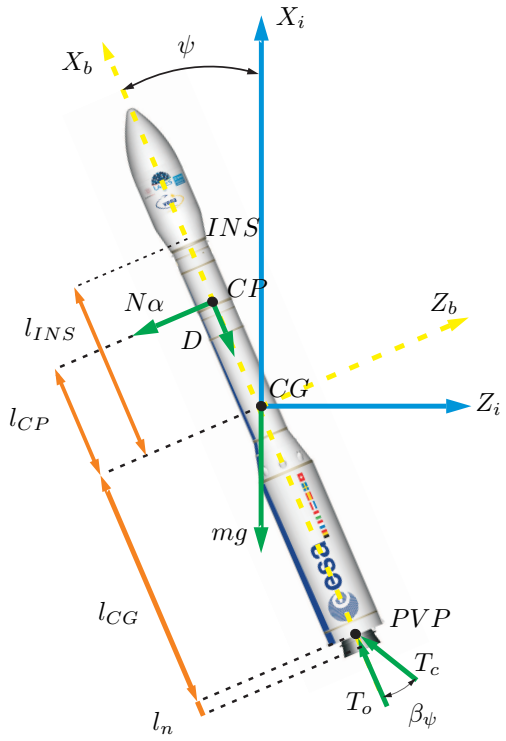


Figure 1: VEGA yaw-motion diagram

$$M_r = Nl_{CP}(\psi + \frac{\dot{z}}{V} + \frac{v_w}{V} - \frac{\dot{\psi}}{V}l_{CP}) - T_c\beta_\psi l_{CG} \tag{4}$$

where T is the total thrust force, which is composed of the gimballed and ungimballed thrust forces, T_c and T_o respectively. D represents the aerodynamic drag force and N the aerodynamic normal force which is given by $N = QS_{ref}CN_\alpha$, where Q is the dynamic pressure, S_{ref} is the launcher reference area and CN_α is the lift gradient. l_{CP} is the distance from the center of gravity (CG) to the aerodynamic center of pressure (CP) and l_{CG} is the distance from CG to the nozzle pivot point (PVP). ψ is the yaw attitude angle, α is the angle of attack, β_ψ is the actuator deflection in the yaw plane, V is the launch vehicle speed, $\dot{z}$ is the vehicle lateral drift rate and v_w is the wind speed.
Similarly, the lateral force F_n and moment M_n due to the nozzle dynamics are given by:

$$F_n = -m_n l_n \ddot{\beta}_\psi \tag{5}$$

$$M_n = -(m_n l_n l_{CG} + I_n)\ddot{\beta}_\psi \tag{6}$$

where m_n is the nozzle mass, l_n is the distance from the nozzle center of gravity to the PVP. The moment of inertia of the nozzle engine about the PVP, I_n, is given by $I_n = I_o + m_n l_n^2$, where I_o is the moment of inertia of the nozzle engine about its center of gravity.

Finally, the sensed values are defined at the node location of the inertial navigation system (INS):

$$\psi_{INS} = \psi \tag{7}$$
$$z_{INS} = z - l_{INS}\psi \tag{8}$$
$$\dot{z}_{INS} = \dot{z} - l_{INS}\dot{\psi} \tag{9}$$

For analysis and design purposes, all the aforementioned relevant dynamics are generally expressed using a state-space formulation, as follows:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_1 & a_3 & a_2 \\ 0 & 0 & 0 & 1 \\ 0 & a_4 & a_6 & a_5 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ a_p & k_2 & -a_1 \\ 0 & 0 & 0 \\ k_1 & k_3 & -a_4 \end{bmatrix} \begin{bmatrix} \beta_\psi \\ \ddot{\beta}_\psi \\ v_w \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \psi_{INS} \\ z_{INS} \\ \dot{z}_{INS} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & -l_{INS} & 0 \\ 0 & 1 & 0 & -l_{INS} \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_\psi \\ \ddot{\beta}_\psi \\ v_w \end{bmatrix} \tag{11}$$

The previous state-space model is composed of four rigid-body states (z,$\dot{z}$, ψ, $\dot{\psi}$); three inputs (β_ψ, $\ddot{\beta}_\psi$, v_w) and three outputs at the inertial navigation system (INS) position (ψ_{INS}, z_{INS}, $\dot{z}_{INS}$). The matrix coefficients are defined as follows:

$$\begin{aligned} a_1 &= \tfrac{-N}{mV}; & a_2 &= -a_1 l_{CP}; & a_3 &= \tfrac{-(T-D)}{m} + a_1 V; \\ a_4 &= \tfrac{N}{J_{yy}V} l_{CP}; & a_5 &= -a_4 l_{CP}; & a_6 &= a_4 V; \\ k_1 &= \tfrac{-T_c}{J_{yy}} l_{CG}; & a_p &= -\tfrac{T_c}{m}; & & \\ k_2 &= -\tfrac{m_n}{m} l_n; & k_3 &= \tfrac{-1}{J_{yy}}(m_n l_n l_{CG} + I_n); & & \end{aligned} \tag{12}$$

2.3 VEGA controller structure

The VEGA TVC control architecture for each channel (pitch and yaw) [2] is based on a PD controller to stabilize the launcher's attitude, a lateral control feedback to reduce the angle of attack and to minimise the drift of the vehicle and a set of bending filters $H_\#$ (with $\# = 1 - 4$) to attenuate the bending modes (see Figure 2).
As mentioned in Section 1, the controller for the atmospheric phase is designed using a GS approach. First, a controller is design at every operational design point considering the launch vehicle model is a LTI model. Then, the controller gains and filters are scheduled based on some parameter (i.e. time, non-gravitational velocity, etc). Since in this work the flexible-body motion is not considered, the bending filters will not be implemented. The control law is defined in equation 13.

$$\beta_{\psi_c}(s,t) = \Delta_\psi(t)[K_{p_\psi}(t) + sK_{d\psi}(t)] + K_z(t)z(t) + K_{\dot{z}}(t)\dot{z}(t) \tag{13}$$

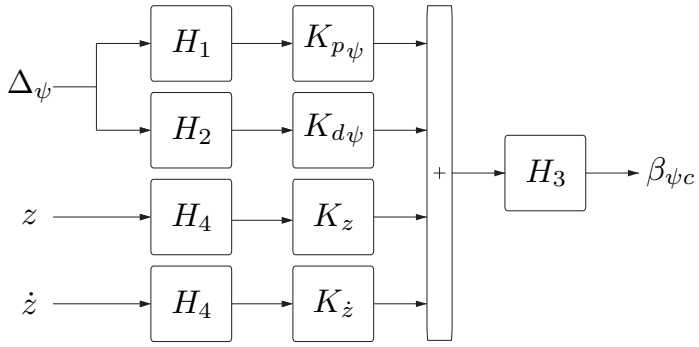


Figure 2: VEGA TVC Control Architecture

2.4 VEGA requirements

The TVC control system must satisfy very demanding stability and performance requirements. The most relevant specifications for the atmospheric phase are listed in Table 1 [6]. Some of the exact values of the requirements are not shown for confidential reasons.

	Requirements	**Metrics**	**Bounds**
Nominal Stability Requirements	Rigid-body margins	LF-GM DM HF-GM	$\geq$ 6 dB $\geq$ 100 ms $\leq$ - 6 dB
Performance Requirements	Load Requirements	$Q\alpha$	$< Q\alpha$ envelope
	Lateral Control Requirements	z $\dot{z}$	$< XX$ m $< YY$ m/s
	Actuation Requirements	β_ψ	$< ZZ^o$

Table 1: VEGA stability and performance requirements for the atmospheric phase

The stability requirements are based on the classical stability margins. In this case, only rigid-body margins are considered: low-frequency gain margin (LF-GM), delay margin (DM) and high-frequency gain margin (HF-GM). Those margins are assessed in the frequency domain through Nichols plots in a SISO approach (pitch and yaw channels are considered decoupled). Since uncertainties are not considered in this work, only nominal stability requirements are provided.
On the other hand, performance requirements must be verified via time-domain Monte-Carlo simulations using the nonlinear high-fidelity simulator VEGACONTROL. For further details on this simulator, the reader is referred to [17]. The different performance metrics must remain below given bounds in the face of parameter dispersion and disturbances such as noise and wind. The loads requirement is expressed as the product of the dynamic pressure and the angle of attack, $Q\alpha$ (see equation 14). To maintain the aerodynamic loads low, $Q\alpha$ must be below a given profile versus Mach.

$$Q\alpha = Q(\psi + \frac{\dot{z}}{V} - \frac{v_w}{V}) \tag{14}$$

Furthermore, the lateral displacement with respect to the reference trajectory frame shall be limited in the atmospheric phase (both in position z and velocity $\dot{z}$). Note that load alleviation and lateral control are opposite strategies, since the load alleviation will change the launcher trajectory to reduce the aerodynamic loads on the vehicle. Finally, the actuator effort shall be also limited to avoid saturation and reduce the TVC consumption.

3 STRUCTURED $\mathcal{H}_\infty$ DESIGN

In this section, the Structured $\mathcal{H}_\infty$ synthesis is used to recover the controller that was used in the VEGA vv05 flight. First, the design framework is introduced and the design for a single flight instant is explained in detail. Then, the evolution of the weighting functions used for the different linear design points is analysed. Finally, the structured $\mathcal{H}_\infty$ design is implemented and validated in a nonlinear high-fidelity simulator.

3.1 Problem formulation

The design model used in this work is based on the closed-loop shown in Figure 3. Note that the real system has only one command, the yaw attitude command ψ_c. The drift and drift-rate commands, z_c and $\dot{z}_c$ respectively, are added to account for the interactions between the attitude and the lateral control channels in the design. In addition, a noise input has been implemented to model sensor errors. The diagram blocks K, G_τ, G_{TVC}, G_{wind} and G_{LV} are described below.

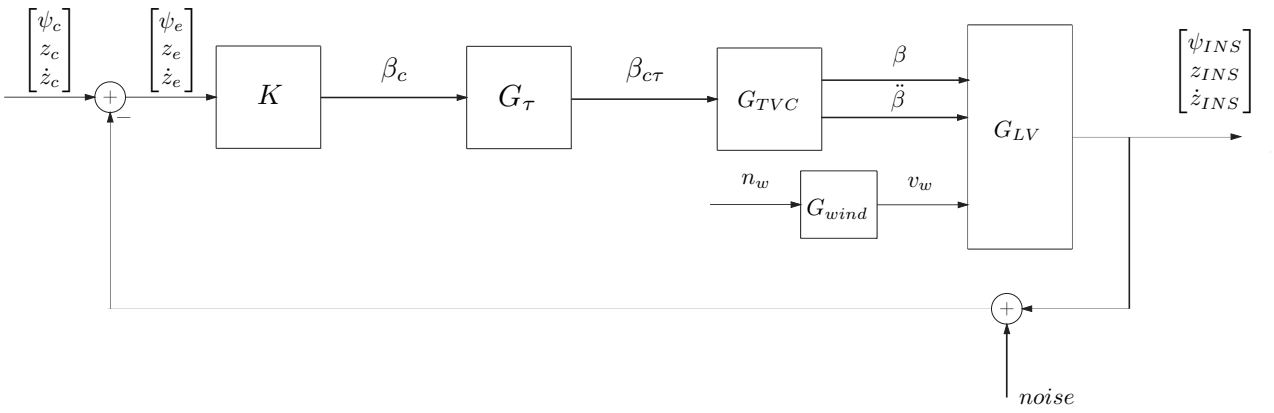


Figure 3: Closed-loop diagram for design

The closed-loop used for design consists of four main blocks:

1. Launch vehicle model (G_{LV})
   This block represents the model described in Section 2.2.

2. Controller (K)
   The controller K is based on the TVC controller structure described in Section 2.3.

3. TVC actuator (G_{TVC})
   This model is derived to fit the actuator dynamics obtained from hardware-in-the-loop simulations. The acceleration of the actuator deflection $\ddot{\beta}_{\psi}$ is also included to account for nozzle dynamics. The details of this model can be found in reference [6].

4. Delay (G_{τ})
   The delay originated by the digital processing of the on-board computers and the actuators is modelled by a 2^{nd} order Padé approximation. In this work, a delay of 39 ms is implemented.

5. Wind generator (G_{wind})
   For an adequate design it is key to introduce wind models that represent the wind profiles that the launcher will encounter in the real flight. In this study, following the criteria found in [18], the wind disturbance input is modelled by coloring white noise n_w through a Dryden filter with the following frequency response function:

$$G_{wind}(s,h) = \frac{\sqrt{\frac{2}{\pi}\frac{V(h)-v_{wp}(h)}{L(h)}\sigma^2(h)}}{s + \frac{V(h)-v_{wp}(h)}{L(h)}} \tag{15}$$

   where the values of the turbulence length scales $L(h)$ and standard deviations $\sigma(h)$ are given in tables in [18] for light, moderate and severe turbulence versus altitude. In this work, moderate turbulences are considered.

   Finally, $v_{wp}(h)$ is computed using the build-up wind speed profile envelope described in equation 16. This envelope is defined for the first 20 Km of altitude, which is the altitude span where the wind disturbance plays an important role, particularly for the $Q\alpha$ performance requirement that depends directly on the wind (see equation 14).

$$v_{wp}(h) = \begin{cases} 10A[(\frac{h}{H_l})^{0.9} - 0.9\frac{h}{H_l}] & \text{for} \quad 0 \leq h < H_l \\ A & \text{for} \quad H_l \leq h \leq H_f - H_u \\ \frac{A}{2}[1 - cos(\frac{\pi}{H_u}(h - H_f))] & \text{for} \quad H_f - H_u < h \leq H_f \end{cases} \tag{16}$$

   where $H_f = 20000$ m, $H_l = 2000$ m, $H_u = 2500$ m and $A = 14$ m/s.

The next step in the approach is to express the control problem in the standard $\mathcal{H}_\infty$ formulation as shown in Figure 4. This formulation allows to consider the full closed-loop system, accounting for all the interactions between channels. P is the generalised plant model with the commands, wind disturbance and noise channel as exogenous inputs and the main signals to be controlled as exogenous outputs (i.e. assuming as endogenous inputs the inputs/output to the controller K).
The output signals of P have been chosen to cope with all the requirements presented in Table 1. Those specifications are introduced in the design process through input and output weighting functions scaling P. Tracking objectives and stability requirements are imposed on ψ_e and ψ_{INS} by W_ψ. W_z and $W_{\dot{z}}$ addresses the lateral control objectives over z_{INS} and $\dot{z}_{INS}$ respectively. $W_{Q\alpha}$ adds constraints to satisfy the load requirements and finally the actuation performance is limited by W_β.

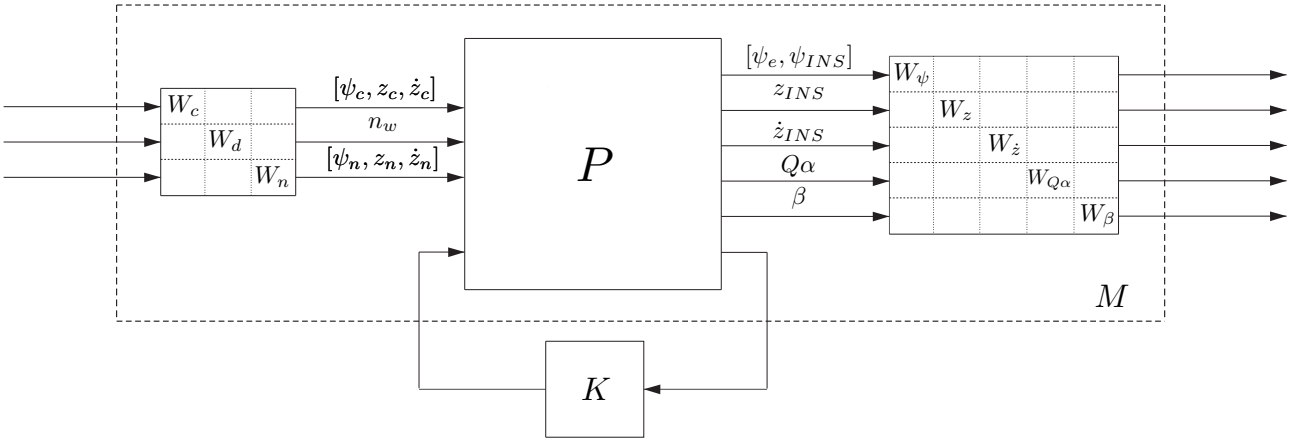


Figure 4: Standard $\mathcal{H}_\infty$ interconnection

The structured $\mathcal{H}_\infty$ optimisation consists of finding the controller K which minimises the $\mathcal{H}_\infty$ norm of the following cost function:

$$\min_K ||\mathcal{F}_l(M, K)||_\infty \tag{17}$$

where $\mathcal{F}_l$ denotes the lower Linear Fractional Transformation (LFT), M is the augmented closed-loop (including the weighting functions) and K is the tunable structured controller, which is of the form:

$$K = [K_p + \frac{s}{T_f s + 1} K_d \quad K_z \quad K_{\dot{z}}] \tag{18}$$

The controller structure defined in equation 18 does not include the bending filters. It should be remarked that for ease of design the derivative bending filter H_2, which performs a derivative action, has been replaced by a first-order filter. K is defined using the Matlab library of tunable blocks *'ltiblock'* as follows:

```
% Attitude tunable controller
Kpsi          = ltiblock('Kpsi','PD');
Kpsi.Tf.Value = 0.005;
Kpsi.Tf.Free  = false;
% Lateral tunable controller
Klat = [ltiblock.pid('Kz','P') ltiblock.pid('Kzdot','P')];
% Full tunable controller
K = [Kpsi Klat];
```

Note that the pseudo-derivative term T_f has been fixed and therefore will not be tuned. This allows to focus the optimisation on the rigid-body controller gains.

As mentioned before, the atmospheric TVC control design is carried out at several operational points along the flight. In this paper, designs are performed every 10 seconds between the flight instants $t = 20s$ and $t = 90s$. Note that this interval does not cover events such as lift off (approx. $t = 4s$) and tail-off separation (approx. $t = 97s$), which have different controller structure and requirements. In this work, the weighting functions are shaped to recover the baseline controller at every operational point design. Next, the weighting functions for the design at a particular flight instant are described.

3.2 Structured $\mathcal{H}_\infty$ linear point design

In this section, the weighting functions used for the design at a particular flight instant (t=50s) are described. It is relevant to mention here that the structured $\mathcal{H}_\infty$ synthesis is a non-convex and non-smooth optimiser, and thus there might not be a unique set of weighting functions which can recover the baseline controller.
Proper scaling of the input and output variables is key for a good control design, particularly working with multiple-input multiple-output (MIMO) systems. It is also important to use the same units at both sides to improve the conditioning of the design problem. In this work, all scaled variables representing angles are expressed in degrees.
Unlike $\mathcal{H}_\infty$ synthesis, in which the order of the weighting functions is generally conserved low to avoid high-order designs, when using the fixed-order $\mathcal{H}_\infty$ synthesis there is no restriction on the order of the weights. Nevertheless, for ease of tuning and to facilitate the design process, constant and first order weighting functions are used.
In this work, the design weights are directly related to physical properties of the launch vehicle and specifications. This helps to improve the understanding of how to express the system requirements in the frequency domain. The main guidelines followed are listed next:

- The input weighting functions are chosen to scale the closed-loop dynamics at the input side with respect to their expected variation.
- W_ψ enforces tracking and stability requirements. On the one hand, the maximum peak sensitivity function ψ_e must be kept small to assure good stability margins. Recall that the peak of the sensitivity function directly yields a lower bound on the classical stability gain margin (GM) and phase margin (PM) through the following relations [3]:

$$GM \geq \frac{||S(j\omega)||_\infty}{||S(j\omega)||_\infty - 1} \qquad (19) \qquad\qquad PM \geq 2\arcsin\Big(\frac{1}{2||S(j\omega)||_\infty}\Big) \qquad (20)$$

  On the other hand, the complementary sensitivity function ψ_{INS} shall be bounded with a low-pass filter to limit the closed-loop bandwidth and to minimise the noise contribution. This bandwidth should be sufficiently high to have an adequate tracking but low enough to avoid interactions with the first bending mode.
- W_z and $W_{\dot{z}}$ shall correspond to the maximum drift and drift rate output expected values.
- $W_{Q\alpha}$ uses the inverse of a user-defined constraint on the angle of attack.
- W_β uses the inverse of the maximum actuator deflection.

The commanded input matrix W_c considers a maximum attitude angle command ψ_c of 1 deg:

$$W_c = \begin{bmatrix} \pi/180 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

The input disturbance weight W_d is defined to account for 3 standard deviations of the unitary white noise input n_w (see Figure 3). This ensures that 99,73 % of the wind levels will be considered in the Gaussian process described by the Dryden filter G_{wind}. W_d is given by:

$$W_d = 3 \tag{22}$$

The input noise weight W_n presented in equation 23 models the sensor noise of each feedback measurement. Firstly, a noise level of 0.02 deg is expected from the IMU sensor used by VEGA. For the lateral deviation measurements, the estimated noise levels provided by the guidance module are 0.01m for the drift and 0.001m/s for the drift-rate.

$$W_n = \begin{bmatrix} 0.02\frac{\pi}{180} & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \tag{23}$$

The output weighting functions impose the system requirements on the design optimisation. In addition, they also scale the closed-loop dynamics at the output side (similarly, all the output angle variables are expressed in degrees).
The initial approach for the selection of the output weighting functions is to shape first the design weights with a certain margin over the baseline frequency responses to give freedom to the optimiser. Then the weight selection becomes an iterative process in which the weights are tuned to recover the baseline controller.
W_ψ establishes the tracking objectives and stability requirements in equation 24. ${W_\psi}^{-1}$ bounds the classical sensitivity and complementary sensitivity functions of the yaw attitude channel, ψ_e and ψ_{INS} respectively. On the one hand, ψ_e is limited by ${W_{\psi_e}}^{-1}$, which is a constant weighting function of 2.5 dB, assuring good stability margins.
On the other hand, ψ_{INS} is weighted by ${W_{\psi_{INS}}}^{-1}$, which is a low-pass filter that limits the closed-loop bandwidth. ${W_{\psi_{INS}}}^{-1}$ presents a low-frequency gain around 8dB, a high-frequency gain of -80dB to force the roll-off at high frequencies and a crossover frequency of 15 rad/s.

$$W_\psi = \frac{\pi}{180}\begin{bmatrix} W_{\psi_e} & 0 \\ 0 & W_{\psi_{INS}} \end{bmatrix} = \frac{\pi}{180}\begin{bmatrix} 0.75 & 0 \\ 0 & \frac{s+5.972}{0.0001s+15} \end{bmatrix} \tag{24}$$

W_z and $W_{\dot{z}}$ enforce the lateral control requirements on the design process. ${W_z}^{-1}$ and ${W_{\dot{z}}}^{-1}$ shall correspond to the maximum drift and drift rate output values. For this design, the maximum drift allowed at $t = 50s$ is 167m (see equation 25) and the maximum drift rate is 8.5m/s (see equation 26).

$$W_z = 1/167 \tag{25}$$

$$W_{\dot{z}} = 1/8.5 \tag{26}$$

The load requirements are set through the weighting function $W_{Q\alpha}$. In this case, the inverse of $W_{Q\alpha}$ puts a constraint of 3 degrees on the angle of attack α, ensuring that $Q\alpha$ is maintained below the safety envelope. $W_{Q\alpha}$ is given by:

$$W_{Q\alpha} = \frac{\pi}{180} Q \frac{1}{3} \tag{27}$$

Finally, W_β imposes the actuation requirements. Similarly, ${W_\beta}^{-1}$ shall refer to the maximum actuator deflection. However, the baseline actuation frequency response has much higher components at high frequencies, which results in the following constant weighting function:

$$W_\beta = \frac{\pi}{180} \frac{1}{35} \tag{28}$$

Although not implemented in this work, it should be mentioned that other requirements, such as actuation rate or angular acceleration, can also be considered in the design.
Using the weighting functions described above and the design interconnection defined in Section 3.1, a structured $\mathcal{H}_\infty$ controller is obtained. To validate the design, Figure 5 compares the magnitudes of the closed-loop transfer functions using the baseline controller (in solid black) and the fixed-order $\mathcal{H}_\infty$ design (in dashed red). Figure 5 shows the frequency responses from four inputs (commands and wind disturbance) to the six output weighted variables. Note that the noise input channels are not displayed for ease of visualization. In addition, this plot also illustrates in green the inverse of the output weighting functions used for the design.
Looking at Figure 5, it can be seen that both frequency responses (baseline and fixed-order $\mathcal{H}_\infty$ design) match for every channel. This implies that the baseline controller at $t = 50s$ is successfully recovered using the structured $\mathcal{H}_\infty$ synthesis.

3.3 TVC control atmospheric design

The same design procedure is repeated for the rest of the linear design points along the atmospheric flight. In total, 8 structured $\mathcal{H}_\infty$ controllers were synthesized between the flight instants $t = 20s$ and $t = 90s$.
Initially, different input weighting functions were tried for the design of the other points, but for ease of design and based on the results it was clear that fixing them and modifying only the output weighting functions was sufficient. Therefore, the same input weighting functions presented in equations 21-23 were used for the 8 stationary designs. Recall that due to the wide dynamic variation of the mission, the design objectives change along the atmospheric phase, and therefore also the output weights, which impose the system requirements on the design. This change can be seen in Figure 6, where the evolution of the inverse of some output weighting functions is depicted. Note that the values in the y axis are not provided for confidential reasons.
Looking at the $W_{Q\alpha}^{-1}$ plot in Figure 6, it can be observed that the inverse of $W_{Q\alpha}$ evolves inversely proportional to the dynamic pressure Q. Furthermore, the actuation requirement decreases with time.

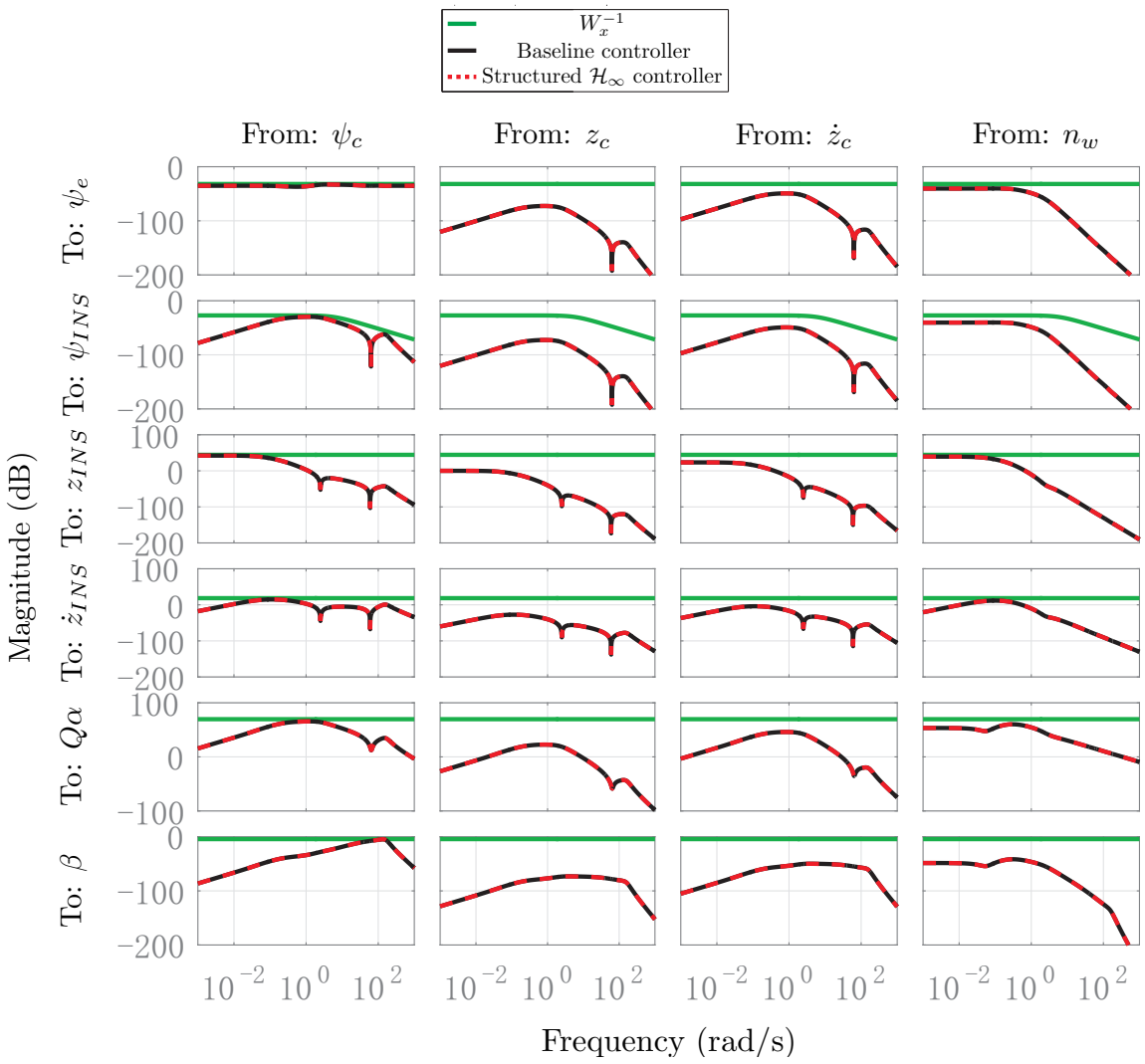


Figure 5: Magnitude Bode plots for the closed-loop transfer functions

This behaviour was expected because, since the controllability parameter k_1 increases with time, the launch vehicle requires less TVC actuation effort to generate the same torque motion. Finally, ${W_z}^{-1}$ and ${W_{\dot{z}}}^{-1}$ follow a similar trend. Two different strategies are differentiated: from t=20 to t=70s, the drift and drift-rate are controlled to satisfy the lateral control specifications, loosening slightly the deviation constraints over the maximum dynamic pressure region (around t=50s) in favour of reducing the aerodynamic loads. At t=80s and t=90s, the controller allows a large drift in preparation for the first stage separation.

Although not shown in Figure 6, W_ψ follows the same strategy explained in Section 3.2. A constant weight for ${W_{\psi_e}}^{-1}$ to constraint the sensitivity function of the attitude channel and a low-pass filter ${W_{\psi_{INS}}}^{-1}$ to limit the closed-loop bandwidth. Note that this bandwidth is higher around the maximum dynamic pressure, which makes the control task more challenging because of flexible motion couplings.

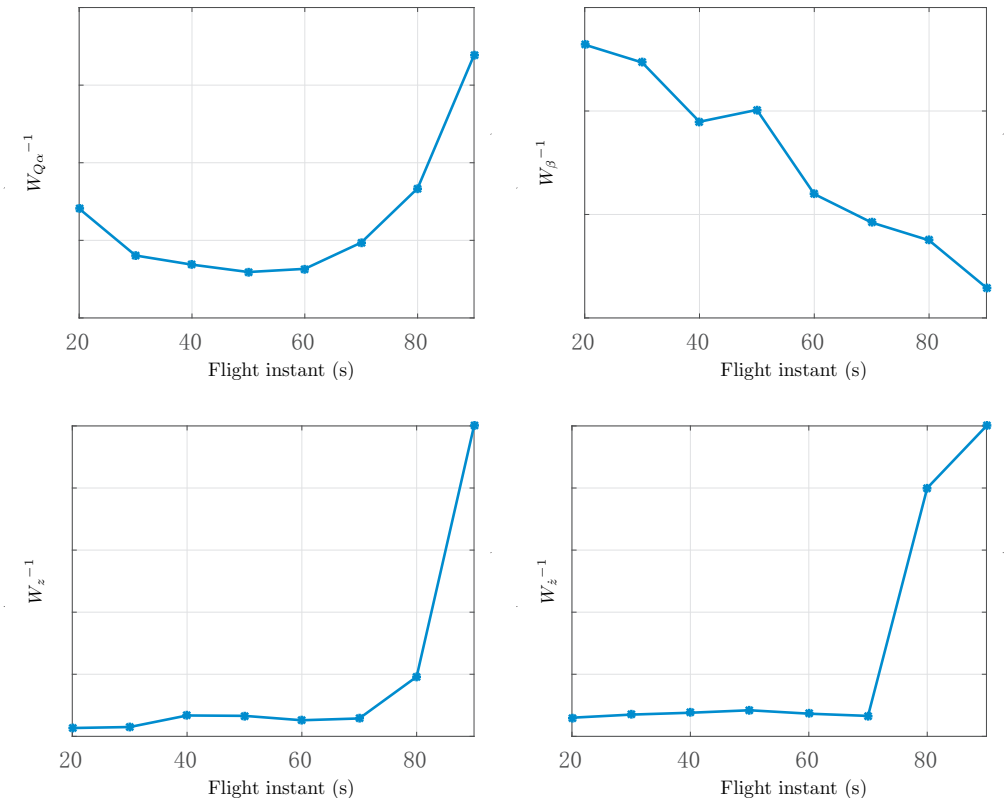
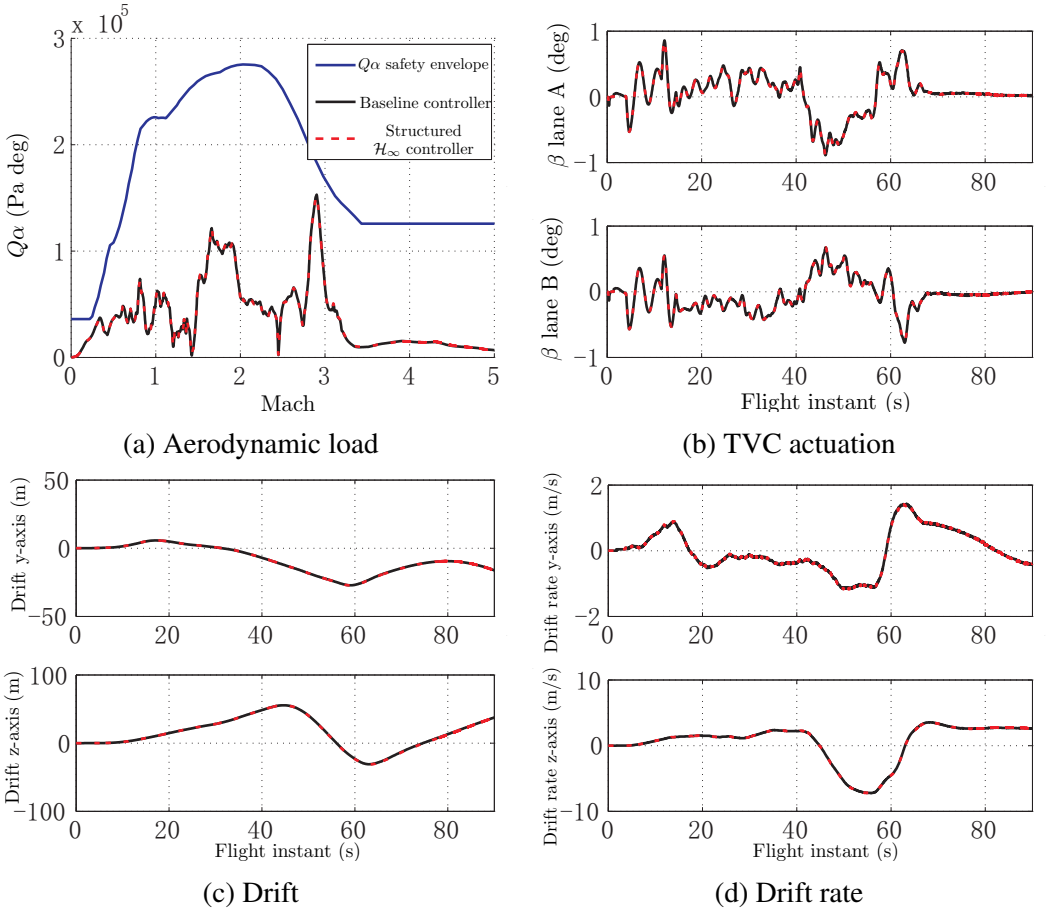


Figure 6: Time evolution of the output weighting functions

It should be reminded that since the structured $\mathcal{H}_\infty$ synthesis is a non-convex optimisation technique, there might not be an unique set of weighting functions which can recover the baseline controller. Despite this fact, the previous analysis provides a good insight of the baseline controller design strategies.

3.4 Nonlinear simulation

Finally, the 8 structured $\mathcal{H}_\infty$ controllers are implemented and validated in VEGACONTROL, the nonlinear 6 degrees-of-freedom simulator for the VEGA launcher. The same controller scheduling rule for the baseline, as implemented in the VEGACONTROL, was used.
In this section, some VEGACONTROL variables (the load indicator $Q\alpha$, drift and drift-rate responses and the TVC deflections) are compared in the time domain in Figure 7 using two different controllers: the baseline controller (in solid black) and the structured $\mathcal{H}_\infty$ design (in dashed red). The simulation is performed for nominal conditions. Figure 7 clearly illustrates that the outcomes for both controllers match.
The same comparison was performed for dispersed conditions in a Monte-Carlo campaign, providing the same outcome for both cases, the baseline and the structured $\mathcal{H}_\infty$ design. The results of this Monte-Carlo analysis are not shown in this article due to space limitations.
Therefore, it can be concluded that the structured $\mathcal{H}_\infty$ design successfully retrieves the baseline controller behaviour for the atmospheric phase.



(a) Aerodynamic load

(b) TVC actuation

(c) Drift

(d) Drift rate

Figure 7: Nominal VEGA VV05 flight responses

4 CONCLUSIONS

In this paper, the structured $\mathcal{H}_\infty$ synthesis technique is applied to recover the baseline controller of the atmospheric phase VEGA launch vehicle attitude control system. The results clearly show that it is possible to recover, using simple weighting functions, the behaviour of the classically, and designed in ad-hoc semi-automated manner, VEGA controller but in a more methodological manner. It has also been shown that when analysing the evolution of the weighting functions then a good overview of the controller design strategies can be extracted.

This is quite promising as it shows a way to exploit, and build upon, the legacy know-how in launcher control design while addressing current shortcomings of the industrial approach. In addition, this work paves the way towards a robust control design framework, which is more suitable for multi-variable control problems and provides better robustness and performance capabilities, as well as powerful robust analysis tools. More recent work of the authors have explored and shown the potentials for improvement by using the structured $\mathcal{H}_\infty$ synthesis technique while keeping the same controller structure.

ACKNOWLEDGMENT

This work was jointly funded by ESA/ELV through the Networking/Partnering Initiative contract No. 4000114460/15/NL/MH/ats. The first author is also the recipient of a Doctoral Training Partnership award by the Engineering and Physical Sciences Research Council (EPSRC). The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency.

REFERENCES

[1] W. J. Rugh and J. S. Shamma, "Survey research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, Oct. 2000.

[2] C. Roux and I. Cruciani, "Scheduling schems and control law robustness in atmospheric flight of vega," in *Proceedings of the International ESA Conference on Guidance, Navigation and Control Systems*, 2008.

[3] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. John Wiley & Sons, 2005.

[4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[5] A. Marcos, S. Bennani, C. Roux, and M. Valli, "Uncertainty modeling and robust analysis of atmospheric launchers: Incremental steps for industrial transfer," *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 426 – 431, 2015.

[6] 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, no. 6, pp. 1342 – 1355, 2016.

[7] P. Gahinet and P. Apkarian, "Structured $\mathcal{H}_\infty$ Synthesis in MATLAB," in *Proceedings of the* 18^{th} *World Congress of the International Federation of Automatic Control (IFAC)*, vol. 18, no. 1, August 2011, pp. 1435–1440.

[8] A. Falcoz, C. Pittet, S. Bennani, A. Guignard, C. Bayart, and B. Frapard, "Systematic design methods of robust and structured controllers for satellites," *CEAS Space Journal*, vol. 7, no. 3, pp. 319–334, 2015.

[9] C. Pittet and P. Prieur, *Structured Accelero-Stellar Estimator for Microscope Drag-Free Mission*. Springer International Publishing, 2015, pp. 591–604.

[10] A. Marcos and M. Sato, "Flight testing of an structured h-infinity controller: an eu-japan collaborative experience," in *Proceedings of the* 1^{st} *IEEE Conference on Control Technology and Applications (CCTA)*, August 2017, pp. 1590–1595.

[11] M. Knoblauch, D. Saussié, and C. Bérard, "Structured H_∞ Control for a Launch Vehicle," in *Proceedings of the American Control Conference (ACC)*, June 2012, pp. 967–972.

[12] D. Saussié, Q. Barbès, and C. Bérard, "Self-Scheduled and Structured H_∞ Synthesis : A Launch Vehicle Application," in *Proceedings of the American Control Conference (ACC)*, June 2013, pp. 1590–1595.

[13] M. Ganet-Schoeller, "Towards structured $\mathcal{H}_\infty$ synthesis for flexible launcher," in *Proceedings of the 3^{rd} CEAS EuroGNC, Specialist Conference on Guidance, Navigation & Control*, April 2015.

[14] D. Navarro-Tapia, A. Marcos, S. Bennani, and C. Roux, "Structured $\mathcal{H}_\infty$ Control Based on Classical Control Parameters for the VEGA Launch Vehicle," in *Proceedings of the IEEE Conference on Control Applications (CCA)*, September 2016.

[15] A. Greensite, *Analysis and Design of Space Vehicle Flight Control Systems. Volume I - Short Period Dynamics*. NASA CR-820, 1967.

[16] J. Orr, M. Johnson, J. Wetherbee, and J. McDuffie, "State Space Implementation of Linear Perturbation Dynamics Equations for Flexible Launch Vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2009.

[17] A. Marcos, P. Rosa, C. Roux, M. Bartolini, and S. Bennani, "An overview of the RFCS project V&V framework: optimization-based and linear tools for worst-case search," *CEAS Space Journal*, vol. 7, no. 2, pp. 303–318, 2015.

[18] C. Andrews, *Terrestrial Environment (Climatic) Criteria Guidelines for Use in Aerospace Vehicle Development*. NASA Technical Memorandum 4511, 1993.