GNC 2014: 9th International ESA Conference on Guidance, Navigation & Control Systems
Porto, Portugal | 2 - 6 June 2014
European Space Agency

AN OVERVIEW OF THE RFCS V&V FRAMEWORK: OPTIMIZATION-BASED AND LINEAR ANALYSIS TOOLS FOR WORST-CASE SEARCH

A. Marcos[1,*], P. Rosa[2], C. Roux[3], M. Bartolini[4], S. Bennani[5].
[1] Deimos Space S.L.U., Madrid, 28760, Spain (e-mail: andres.marcos @bristol.ac.uk).
[2] Deimos Engenharia, Lisbon, 1998-023, Portugal (e-mail: paulo.rosa @deimos.com.pt).
[3] ELV, Rome, 00034, Italy (e-mail: Christophe.Roux@elv.it).
[4] ELV, Rome, 00034, Italy (e-mail: matteo.bartolini@consultant.elv.it).
[5] ESA_ESTEC, Noordwijk, 2201AZ, The Netherlands (e-mail: samir.bennani@esa.int).

ABSTRACT

This article presents the application of nonlinear (simulation-based) and linear (structured singular value) worst-case tools to the VEGA launcher Verification and Validation (V&V) process, during the atmospheric ascent phase. The simulation-based worst-case evaluation is performed by minimizing certain cost functions that are intrinsically aligned with the performance of the launcher using an optimization tool and the high-fidelity nonlinear simulator of the launcher. A numerical sensitivity analysis of these functions is performed prior to the optimization campaigns, in order to select the variables with the largest impact on the criteria. In terms of optimization methods, differential evolution and hybrid differential evolution algorithms are adopted, due to their ability to cope with a large number of optimization variables. The linear analysis uses the structured singular value and a linear fractional transformation model, formed from a subset of the uncertainty and dispersion parameters defined for the nonlinear VEGA system, to identify physically feasible worst-case conditions. This analysis is complementary to traditional Monte-Carlo (MC) campaigns in that it provides an analytically guaranteed existence of worst-case conditions. The simulation-based and the linear analysis approaches adopted in this paper require, in general, only a fraction of the MC runs needed to detect worst-case conditions.

I. INTRODUCTION

The state-of-practice for launcher Verification & Validation (V&V) entails analysis of frequency domain requirements (i.e. stability margins) through a set of predefined vertex cases and, more critically, analysis of time domain requirements through nonlinear simulations in (i) a Monte-Carlo (MC) setting (i.e. probabilistic approach) and (ii) using a set of selected worst-cases (i.e. deterministic approach).

Traditional MC campaigns consist of randomly sampling the uncertain parameters according to statistical distributions and deducing the values of criteria involved in the requirements. The number of simulations depends on specified probability and confidence levels, as described in the several studies available in the literature [6]. Although the MC approach is very practical in showing the design sensitiveness to parametric variations, it has many disadvantages in that it relies on massive amounts of computation without guaranteed proofs on the full parameter space and is questionable for maximum/minimum values analysis. Similarly, vertex approaches, whereby all the corner cases (minimum and maximum along each uncertain parameter direction) are checked, are unfeasible whenever the dimension of the problem is relatively high. This is compensated by introducing engineering knowledge on the system which helps limit the number of corner cases to be examined. Nonetheless, the analysis is very limited and obviates altogether parameter combinations away from the extremes.

To overcome these limitations, many advances have been explored in the field of aeronautical GNC V&V (cf. [5], [3], [8]). One of the venues explored, which has proved very successful, has been the use of advanced optimization-based worst-case search algorithms (see, for instance, [16], [15], [13], [14]). The main downside of these approaches is the lack of guarantees in finding the worst-case. However, these methods typically can identify performance violation cases quickly and their usage requires only minimal adaptation (to the typical simulation models used in industry, e.g. Simulink/Matlab, Fortran, C-code). The other venue that has been followed in order to provide analytical guarantees is the use of analytical approaches such as the structured singular value μ (see [1], [2], [14]). The downside of μ-analysis is its linear nature, exemplified by its reliance on linear fractional transformation (LFT) models typically arising from linear dynamic systems, and the lack of associated probabilities to the identified worst-cases, which are necessary to provide a quantitative risk value.

In this article, we extend the results presented in [13], [14] to the enhanced VEGA simulator during the first atmospheric phase, which now takes into account a number of additional parameters, including the wind profile and the bending modes of the vehicle. It is shown that both the linear analysis and the simulation-based

[*] Now with University of Bristol, Aerospace Engineering Department, Bristol, BS8 1TR, United Kingdom.

approaches are able to successfully identify numerous performance-violating cases, which are valid not only in the nonlinear simulator used for verification, but also in the official VEGA simulator used for validation.
The organization of the paper is as follows: Section II describes the characteristics of the launcher, as well as the problem at hand. Section III investigates the use of optimization-based worst-case search algorithms, while Section IV presents the μ analysis process adopted. The results obtained with each of the approaches are provided and discussed in Section V. Some final remarks are provided in Section VI.

II. VEGA LAUNCHER & NONLINEAR SIMULATORS

VEGA is the new European Small Launch Vehicle developed under the responsibility of ESA. The prime contractor for the launch vehicle is ELV. The launcher successfully performed a maiden launch at the beginning of 2012 from the Centre Spatial Guyanais in Kourou, a 2nd flight in May 2013, and a 3rd flight in April 2014.
VEGA follows a four-stage approach formed by 3 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 four stages are controlled via a thrust vectoring system (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.
The official non-real time high-fidelity nonlinear simulator used in the VEGA program is called VEGAMATH. This simulator is characterized by:

- High-fidelity 6 Degrees-of-Freedom motion
- Tail-Wag-Dog effects
- Bending and sloshing modes
- Full external environment (rotating Earth, winds…)
- Disturbances (torques at separation, bias, offsets)
- Nonlinear aerodynamics (incl. aero-elastic effects)
- TVC system (incl. computing delays, backlash, bias)
- Full representative code implementing the actual Guidance, navigation and control (GNC) system
- Propulsion and mass-center-inertia (MCI) properties
- Detailed inertial navigation system (INS)
- Detailed RACS models (thermal & thrust dynamics, filters, quantization and noise…)

The nonlinear simulator used in the RFCS project is called VEGACONTROL and is based on VEGAMATH but tailored to simulate the 1st atmospheric phase (P80). In comparison with [14], the improved VEGACONTROL version used in this work now includes the bending/sloshing modes, the RACS model, wind-scaling profiles, and enhanced correlation of key parameters. This simulator is used by the RFCS consortium to develop and verify their approaches, and the results from these verifications are validated by ELV on the VEGAMATH.
Both VEGACONTROL and VEGAMATH allow modifying the scattering values (uncertainties and dispersions) of up to 125 different uncertain parameters organized into 9 sets (aeroelasticity, aerodynamics, wind, IRS, thrust, MCI, thrust offset & misalignment, atmosphere, separation disturbance). Each scattering variable is represented by a "flag" parameter with values ranging in the interval [-1, 1], with the zero value indicating "nominal".

III. SIMULATION-BASED WCC SEARCH METHODS

In reference [13], a worst-case time-domain optimization toolbox referred to as WCAT [15] was used to identify worst-cases on the previous version of the VEGACONTROL simulator. Using in an iterative manner a number of pre-defined parameters sets, 58 cases that violated one or several performance bounds were identified and verified in the VEGACONTROL and validated in VEGAMATH. The WCAT optimization problem is given by:

$$\begin{aligned} &\min\ C_{\#} \quad where \quad C_{\#} = \left| i_{ref} - i_{\Delta} \right| \\ &s.t.\ -1 < \Delta < +1 \end{aligned} \tag{1}$$

The cost function $C_{\#}$ is defined based on i_{ref}, the reference limit or nominal value of the specific performance index, and i_{Δ}, the actual performance index value obtained in the presence of the uncertainty parameter set Δ used in the simulation [14].
This section adopts the aforementioned WCAT toolbox, to address the problem of detecting worst-cases on the updated version of the VEGACONTROL simulator, which includes additional unknown parameters related to model uncertainty, as previously described. The optimization campaigns are implemented using the Differential Evolution (DE) and Hybrid Differential Evolution (HDE) genetic algorithms, preceded by an approximate sensitivity analysis of each criterion to all and each of the variables.
The methodology adopted can be structured in the following fashion:

- **Step 1 - Sensitivity analysis**: This step is devoted to the analysis of the sensitivity of each cost function, by varying all the parameters around desired points, and finally selecting the variables that have the largest impact on the criteria.
- **Step 2 - Worst-Case assessment**: Using the previously selected variables, worst-case optimization campaigns are performed in this step.
- **Step 3 - Reducing the uncertainty range**: In order to devise worst-case conditions (WCCs) that are more likely to occur, the width of the interval in which the parameters can take value is reduced in this step (e.g. [-1, 1] → [-0.6, 0.6]), followed by a search for WCCs.
- **Step 4 - Comparison with VEGAMATH**: Finally, the results obtained are assessed in VEGAMATH.

The **cost functions** selected for analysis, and that are to be optimized following equation (1) in order to obtain the worst-case scenarios in terms of performance, are as follows:

1. $\mathbf{C_1}$: Qα (dynamic pressure times angle-of-attack, a well-known launcher metric for atmospheric stage)
2. $\mathbf{C_2}$: TVC A Consumption
3. $\mathbf{C_3}$: TVC B Consumption

Criteria C_2 and C_3, however, are typically aligned between them, and thus a single cost function can be considered instead (namely, C_2, C_3, or $C_2 + C_3$).

A. APPROXIMATE SENSITIVITY ANALYSIS

This section provides a sensitivity analysis of the three cost functions considered, with respect to the parameters of the launch vehicle, allowing selecting a subset of variables that are likely to help detect worst-case operating conditions.

As an example, Fig. 1 shows the effect of the 'dTc' parameter (related to burn time) on a set of relevant system variables. In the analysis, 'dTc' is set iteratively to {-1, 0, 1} while all the remaining flags are fixed respectively to {-1, 0, 1} – a total of 9 plots per analysis. The figure shows the results for all the flags set at {-1} and 'dTc' at {-1, 0, 1}. The impact of this parameter on the launcher dynamics is apparent, which indicates that 'dTc' is likely one of the variables to take into account in the search for WCCs.

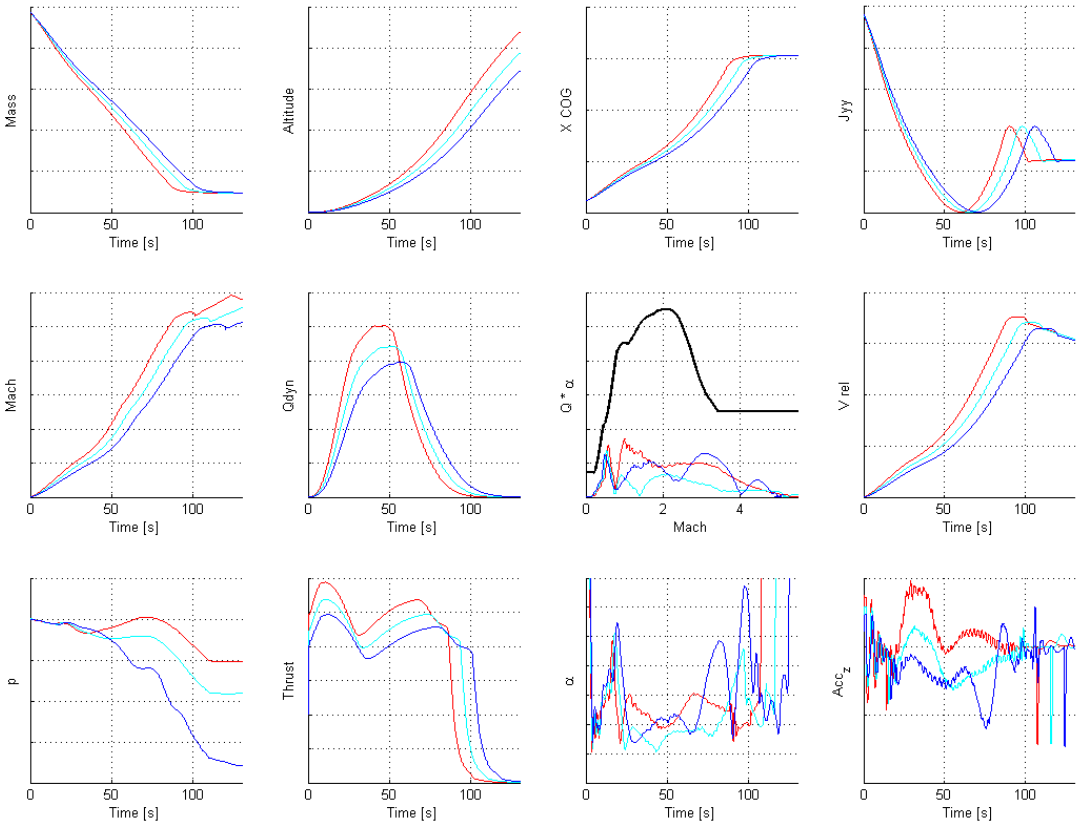

Fig. 1 *Time responses of the system for 'dTc' {-1, 0, 1} (red, cyan, and blue, respectively) and all other flags at {-1}*

In order to perform such an analysis in a more systematic manner, the following algorithm was adopted to compute a numerical approximation of the sensitivity of each criterion to the parameters of the dynamics:

Algorithm 1

```
for each k ∈ {-1, 0, 1}, do
     all flags = k
     for each j ∈ {-1, 0, 1}, do
            variable of interest = j
            run simulation
            c(j) = value of the cost function
     end
     d(k) = variance of sequence (c(-1), c(0), c(1))
end
sensitivity = maximum(d(-1), d(0), d(1))
```

Therefore, each of the 125 variables is iteratively set at {-1, 0, 1}, while all other flags are set at {-1, 0, 1}. For each run, the values of the cost functions are calculated, while normalizing the corresponding variability with respect to the maximum value of each criterion. Thus, this leads to *125×3×3=1125* simulation runs, regardless of the number of cost functions to be evaluated. This algorithm can also be illustrated by means of Fig. 2, where each blue dot corresponds to a simulation run.

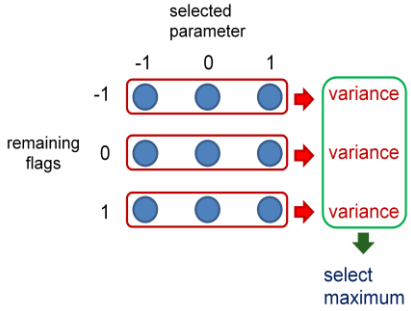

Fig. 2 *Illustration of the behavior of Algorithm 1*

This sensitivity analysis of the cost functions can be extended by varying the flags within different ranges. As illustrated in Fig. 3, if only the extreme values of the intervals are considered, one may not obtain an adequate approximation of the actual sensitivity of each cost functions.

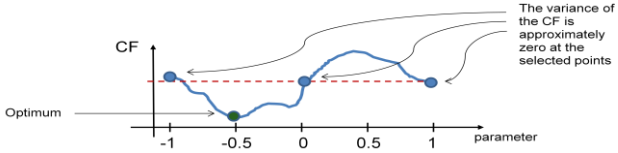

Fig. 3 *Example of a non-convex cost function (CF)*

As an example, we analyze the impact of each parameter on the 3 selected criteria, by considering the variation of all the flags in the following three sets: $S_1 = \{-1, 0, 1\}$; $S_2 = \{-0.6, 0, 0.6\}$; $S_3 = \{-0.3, 0, 0.3\}$. The results obtained are illustrated in Fig. 4. These results lead to a number of conclusions in terms of the variables that should be selected, as well as the expected optimization results to be obtained.

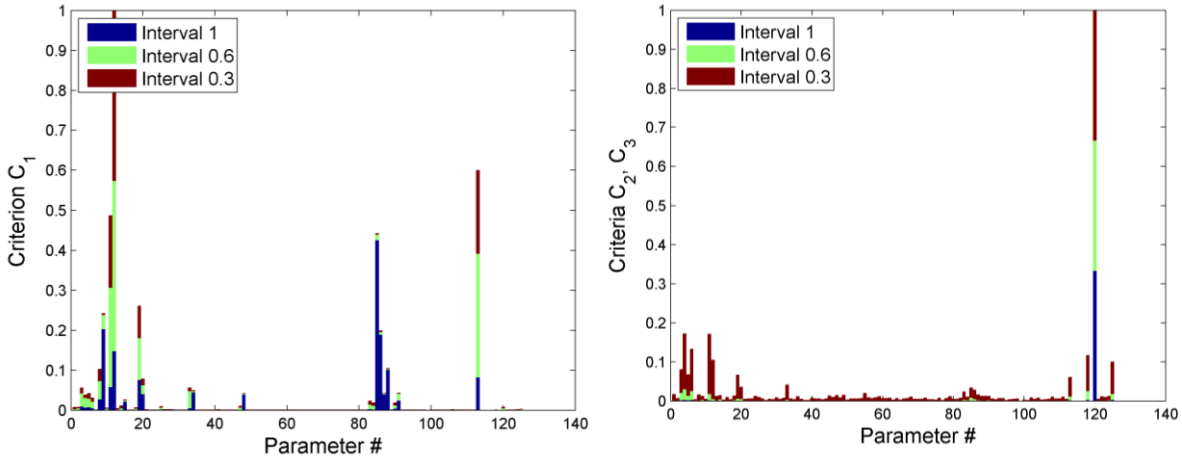

Fig. 4 *Sensitivity of C1 (left), and C2 and C3 (right), to all the parameters*

Regarding C_1, parameters such as #12 (i.e., 'IRSmountingZ') clearly have a significant impact on the cost function, for all of the intervals considered. Other parameters, such as #85 and #86 (i.e., 'TVC_bias_A' and 'TVC_bias_B', respectively), however, only have a pronounced impact on this criterion if the uncertainty interval [-1, 1] is considered. This indicates that one is not expected to obtain any significant benefits from the use of these variables in the optimization procedure, if a small parameter interval is used. Moreover, only about 25 out of 125 parameters seem to have a direct impact on C_1. Thus, a considerably reduced set of variables may be used for optimization purposes.

In terms of TVC consumption, C_1 and C_2, the conclusion is clearly different. Although parameter #120 (i.e., 'flex_freq') is the one with the largest impact on this cost function for all of the uncertainty intervals analyzed, the remaining parameters have a non-negligible contribution, especially if the smallest parameter interval is considered. In other words, perturbing only one variable completely changes the value of the TVC consumption. This fact indicates that the worst-case conditions of the VEGA launcher, for criteria C_2 and C_3, are not likely to be derived using a small subset of parameters.

B. SELECTION OF WORST-CASE PARAMETERS SETS

A total of four parameter sets are considered in this phase, based on the previous analyses and extracted from the 125 uncertain scatterings available in VEGACONTROL and VEGAMATH:

Table 1 *Sets of flags to be optimized*

Set	Flags
Mixed-1	'dTc', 'air_density_scat', 'disp_CN', 'unc_CN', 'disp_Xcp', 'unc_Xcp', 'backlash'
Mixed-2	'h_wind', 'TVC_bias_A', 'TVC_bias_B', 'backlash', 'dTc', 'SRM_roll'
Mixed-3	'TVC_bias_A', 'TVC_bias_B', 'IRSmountingZ', 'SRM_roll', 'backlash', 'unc_Xcp', 'flex_freq'
Mixed-4	'SRM_roll', 'backlash', 'unc_Xcp', 'flex_freq', 'air_density_scat', 'P80dyCOG', 'dTc'

The first two sets were derived for the previous version of VEGACONTROL (see [13]), while

- Mixed-3 contains the variables for which each of the selected criteria are more sensitive, if only the interval [-1, 1] is considered in Algorithm 1, and
- Mixed-4 contains the variables for which each of the selected criteria are more sensitive, using the overall results of Fig. 4, i.e., by using intervals [±0.3], [±0.6], and [±1].

These two sets will be used in the search for WCCs in Section V.

IV. WORST-CASE μ-ANALYSIS AND LFT THEORY

A concept widely used in robust control is the structured singular value μ, which analytically evaluates the robustness of uncertain systems ([4], [1]). It is highlighted that the μ yields worst-case combinations of the uncertain set used in the analysis using a linear frequency-domain framework. In the case of the launcher, this translates into analyzing a set of uncertain linear-time-invariant models obtained at different points in the flight (e.g. at a different time instance). If a worst-case is found at any of these instances then it is verified in the VEGACONTROL nonlinear simulator for the entire flight, and if it still yields a performance violation then validated in the VEGAMATH official nonlinear simulator. Thus, despite μ looking for worst-cases at specific time instances, if a case is found and verified & validated in the nonlinear simulators then the worst-case is valid for the entire flight. In the case that the time instance used for the μ analysis is not be the most representative of the flight, e.g. at the beginning of the vertical flight, then this suggest that the worst-case obtained is independent of the flight region but rather of the internal launcher geometry, MCI or other endogenous property.

A key aspect on the application of μ is the development of a proper LFT model. By proper it is meant a model that captures the critical parametric behavior of the nonlinear system under consideration. In this section, the theory behind μ analysis and LFT modeling is briefly reviewed.

A. STRUCTURED SINGULAR-VALUE ANALYSIS

The structured singular value $\mu_{\Delta}(M)$ of a matrix $M \in C^{n\times n}$ with respect to the uncertain matrix Δ is defined in (1), where $\mu_{\Delta}(M)=0$ if there is no Δ satisfying the determinant condition.

$$\mu_{\Delta}(M) = \frac{1}{\min_{\Delta}(\bar{\sigma}(\Delta) : \det(I - \Delta M) = 0)} \qquad (2)$$

Note that this definition is given in terms of an $\{M,\Delta\}$ model which is an LFT model where Δ is typically norm-bounded $\|\Delta\|_{\infty}\leq 1$ (without loss of generality by scaling of M) for ease of calculation/interpretation. In this manner, if $\mu_{\Delta}(M)\leq 1.0$ then the result guarantees that the analyzed system, represented by the LFT, is robust to the considered uncertainty level.

The structured singular value is a robust stability (RS) analysis but can be used also for robust performance (RP), as this problem can be transformed very straightforward into an RS problem (see [4]).
Since $\mu_{\Delta}(M)$ is difficult to calculate exactly, the algorithms implement upper and lower bound calculations [1]. The upper bound μ_{upper} provides the maximum size perturbation $|\Delta|_{\infty}=1/\mu_{upper}$ for which RS/RP is guaranteed, while the lower bound μ_{lower} guarantees the minimum size perturbation $|\Delta|_{\infty}=1/\mu_{lower}$ for which RS/RP is guaranteed to be violated. Thus, if the bounds are close in magnitude then the conservativeness in the calculation of μ is small, otherwise nothing can be said on the guaranteed robustness of the system for perturbations within $[1/\mu_{upper}, 1/\mu_{lower}]$.

B. LINEAR FRACTIONAL TRANSFORMATION MODELING

An LFT is a representation of a system using two matrix operators, *M=[M11 M12; M21 M22]* and Δ, and a feedback interconnection. The matrix *M* represents the nominal (known) part of the system while Δ contains the unknown, time-varying or uncertain parameters ρ_i. Depending on the feedback interconnection used, there are two possible types of LFTs: lower and upper (see Fig. 5).

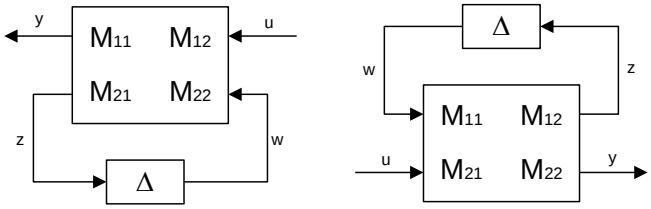

Fig. 5 *Lower and upper LFTs*

The matrix Δ is unrestricted in form (structured or un-structured) or type (nonlinear, time-varying or constant). It is important to note that unstructured uncertainty at component level becomes structured at system level. The selection of the variable set $\rho(t)\in\Delta$ that captures the behavior of the nonlinear system is a task that is not always obvious a priori. Indeed, this step is key to obtain a reliable LFT that will yield relevant and meaningful results and, despite its apparent simplicity, is where most of the LFT modeling effort and ingenuity is focused.
There are several approaches that can be used to obtain a reliable LFT model (see [4], [9], [1], [10], [11], [12] and references therein). The specific approach used in the RFCS project is based on the developments from the last two references, which formalized a modeling methodology to transform a general linear parameter varying model into an LFT representation through the use of symbolic manipulations.

C. FROM EQUATIONS OF MOTION TO LFT

For any launcher vehicle, its yaw or pitch rigid-body motion dynamics is completely described by its attitude (yaw ψ or pitch θ) and linear motion (z or y) in a frame linked to the velocity of the reference trajectory. Under the assumption of a perfect axial-symmetric launch vehicle, the dynamic equations can be expressed in state-space model format as follows:

$$A=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & a_1 & a_3 & a_2\\ 0 & 0 & 0 & 1\\ 0 & a_4 & A_6 & a_5\end{bmatrix};\quad B=\begin{bmatrix}0\\ a_P\\ 0\\ K1\end{bmatrix};$$
$$C=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 1 & 0 & -a_Z\\ 1 & 0 & -a_Z & 0\end{bmatrix};\quad D=\begin{bmatrix}0\\ 0\\ 0\end{bmatrix};\qquad (3)$$

For the yaw rigid-motion, the model has 4 states ($z, \dot{z}, \psi, \dot{\psi}$), 3 outputs ($\Delta\psi, \dot{z}, z$) and 1 input (nozzle deflection angle), where the above {*A, B, C, D*} matrix coefficients are given by:

$$L_\alpha = S_{ref} q_{dyn} C_N \frac{1}{\alpha}; \qquad \mu_a = -L_\alpha (x_{CP} - x_{CoG}) \frac{1}{J_{yy}};$$

$$a_1 = -L_\alpha \frac{1}{m} \frac{1}{V_{rel}}; \qquad a_2 = -a_1 (x_{CP} - x_{CoG});$$

$$a_4 = -\mu_a \frac{1}{V_{rel}}; \qquad a_5 = -a_4 (x_{CP} - x_{CoG}); \qquad (4)$$

$$a_z = x_{INS} - x_{CoG}; \qquad a_P = -T_h \frac{1}{m};$$

$$a_3 = -acc + a_1 V_{rel}; \qquad A_6 = a_4 V_{rel};$$

$$K_1 = -T_h (x_{CoG} - x_{PVP_{ref}}) \frac{1}{J_{yy}};$$

The given variables are defined as: relative velocity V_{rel}, dynamic pressure q_{dyn}, thrust Th, total launch vehicle mass m, total y-axis moment of inertia J_{yy}, normal aerocoefficient C_N, longitudinal acceleration acc, angle of attack α, x-coordinate center of pressure x_{CP} and x-coordinate center of gravity x_{CoG}. All these variables can be represented as depending on time t or on non-gravitational speed, VNG. The launcher dynamic model given by (3)-(4) captures the main characteristics of the 1st atmospheric phase (P80) and is typically used to design the launchers' controllers for this phase.

After careful assessment of the variables above and the scatterings 'flags' in the VEGACONTROL simulator, several nonlinear simulations were performed to determine the 'best' (i.e. most representative of the effects and with least number of parameters) set of parameters for the LFT. Due to space limitations, and the analysis process orientation of this article, no detailed description of the LFT modeling process is given. Nonetheless, in order to provide a basic understanding on the obtained LFT, the selected uncertainty set is given by 6 scatterings 'flags' (Δ={*dTc, air_density_scat, disp_CN, unc_CN, disp_Xcp, unc_Xcp*}) associated to four of the variables in equation (4): combustion time (thru Th), air density (ρ thru q_{dyn}), normal aerodynamic coefficient C_N and center of pressure x-coordinate x_{CP}. These four time-varying variables are introduced in the LFT using linear and bilinear polynomial fits based on the 6 scatterings 'flags' plus symbolic nominal and deviation constants (assigned to M). The final order of the LFT is 31 and it is highlighted that it is a symbolic representation of the system in Eqs. (3)-(4). Also, note that the scattering set correspond to the set mixed-1 from Table 1 (without the 'backlash' scattering)

D. ANALYSIS PROCESS

The analysis process is as follows:

1. Starting from the symbolic LFT model, select the time along the P80 ascent trajectory at which the analysis is to be performed.
2. This will result in a specific VNG at which the symbolic constants in the LFT matrix M are numerically evaluated. Subsequently, this numerical LFT can be model reduced to obtain an LFT order of 22. Note that except for this latter numerical model reduction, the symbolic LFT is fixed and applicable to any launcher in atmospheric phase. Thus, it can be used in an automated or on-board fashion.
3. Close the above LFT system with the TVC controller + actuator + delays loop. The appropriate TVC controller is obtained based on data from ELV and depends on the specific VNG value.
4. Select the proper μ structure for the Δ matrix. The structure can be real, complex or a mix of both ([1]). The advantage is that more efficient algorithms can be used for the last two types. Also in this step, the RS or RP formulation is defined. The difference is that the performance input/output channels for the RP case are closed using a fictitious uncertainty structure of appropriate dimensions, so that an RS problem is obtained.
5. Perform the μ analysis. This is straightforward based on the previous steps.
6. Examine and verify the results. This step serves to ensure that the upper/lower bounds are close but also to determine the validity of the perturbation (i.e. proper size, nature and one that results in actual violation in VEGACONTROL).
7. The results are validated by ELV using the official VEGAMATH simulator.

V. RESULTS

This section describes the results obtained with both methodologies adopted in this work, i.e., optimization-based WCCs search and μ-based linear analysis.

A. SIMULATION-BASED WCC SEARCH RESULTS

Taking into account the results obtained in [13], the Hybrid Differential Evolution (HDE) worst-case optimization algorithm was considered, as this is the technique available in the WCAT toolbox leading to the best results, for the present case.

Based on the analyses in Section III, the selected WCC optimization campaigns are:

- Number of iterations: up to 2500;
- Parameter sets: Mixed-1/2/3/4;
- Cost-functions: C_1, C_2, C_3.

Each of the 4 parameter sets is optimized separately according to each of the 3 cost-functions. Moreover, each optimization is performed by setting all the remaining flags respectively to {-1, 0, +1}.

Since each iterations requires approximately 3.3 seconds in a computer with an Intel Core i5 CPU @ 3.20 GHz, the total time required for the full campaign is: $2500 \times 3 \times 3 \times 4 \times (3.3/3600) \approx 82.5$ hours. Alternatively, each criterion may be computed in a different Matlab session or by using Matlab's Parallel Computing Toolbox, which may significantly speed up the optimization process.

The results obtained are exemplified in Table 2 using the Qα (C_1) cost-function (note that a value smaller than 1 indicates violation of this criterion). It can be observed that clear violations of this criterion were detected, regardless of the set of optimization parameters considered.

Table 2 *Results for 2500 iterations, Qα, HDE. A value smaller than 1 indicates violation of the criterion*

Set	Rest of flags		
	-1	**0**	**1**
Mixed-1	0.7852	0.7662	0.6239
Mixed-2	0.9073	0.9298	0.7822
Mixed-3	0.8847	0.9345	0.8363
Mixed-4	0.6401	0.8033	0.6756

For the TVC A consumption (cost-function C_2), the results obtained are summed up in Table 3. The results for the TVC B (C_3) consumption are omitted, as they follow very closely those of TVC A and for ease of presentation. Unlike the Qα criterion, for the TVC consumption, no violations were detected.

Table 3 *Results for 2500 iterations, TVC A Consumption, HDE*

Set	Rest of flags		
	-1	**0**	**1**
Mixed-1	1.7420	1.7544	1.7889
Mixed-2	1.7736	1.7866	1.7991
Mixed-3	1.7788	1.8015	1.7790
Mixed-4	1.7059	1.7600	1.7391

The simulation results in VEGACONTROL of the corresponding worst-cases obtained by optimizing C_1 and C_2, respectively, are illustrated in Fig. 6 (first and second columns respectively). The violation of Qα is apparent from the plot in the upper left corner of Fig. 6 while no violation of the TVC consumptions was observed (second and third rows) even for the specific case of C_2.

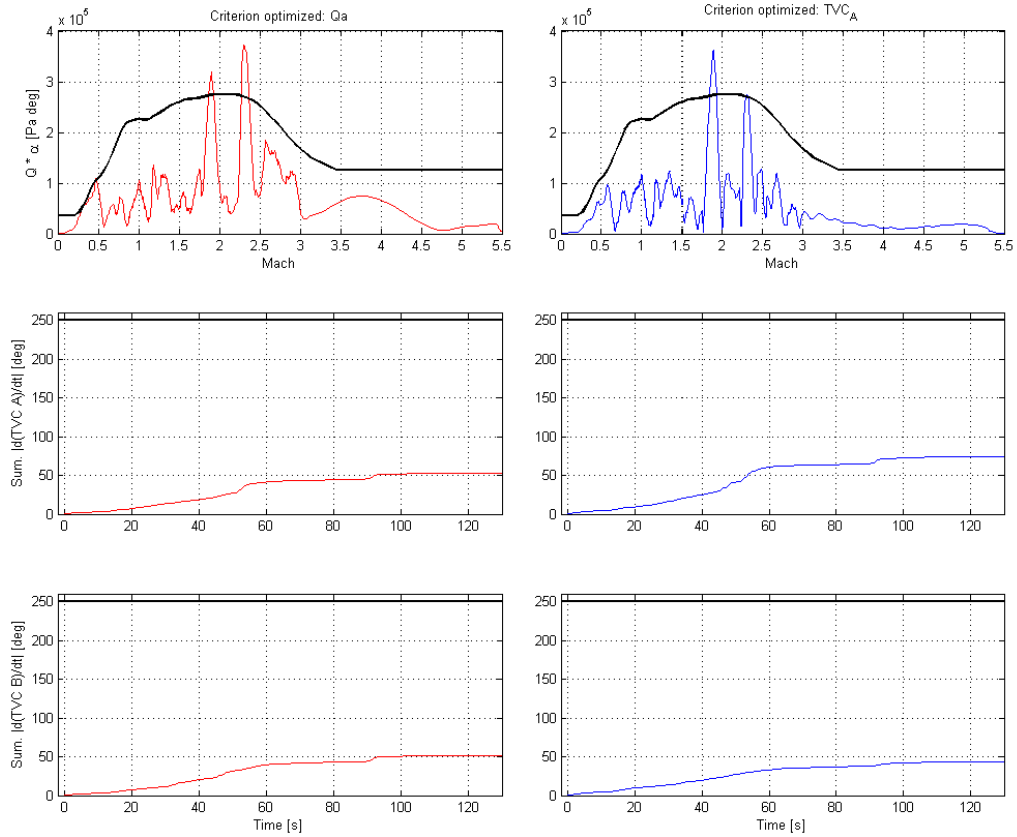

Fig. 6 *WCs obtained by optimizing Qα and TVC consumption. Bold black lines indicate admissible values for each criterion*

UNCERTAINTY RANGE REDUCTION

In this subsection, the width of the allowable range of variables is reduced, in order to obtain WCCs that are more likely to occur in practice. In general, having parameters at the extremes of the interval [-1, 1] is a low probability event, especially if one assumes a unimodal probability distribution of the parameters.
The results obtained are summarized in Table 4 and Table 5, for the cases where the parameters are allowed to range in {-1, 0, 1} and are fixed to {0}, respectively. As in the previous scenario, only violations of the Qα criterion are detected. Sets Mixed-1 and Mixed-4 are the ones that enable the detection of the worst-case conditions. As expected, reducing the uncertainty range also diminishes the severity of the WCC.

Table 4 *Uncertainty range reduction results*

Criteria	[-1, 1]	[-0.66, 0.66]	[-0.6, 0.6]
Qα	0.6223 (Mixed-1)	0.8414 (Mixed-4)	0.8677 (Mixed-4)
TVC A/B	1.5179 (Mixed-4)	1.7930 (Mixed-1)	1.7720 (Mixed-4)

Table 5 *Uncertainty range reduction results, with the remaining flags equal to zero*

Criteria	[-1, 1]	[-0.66, 0.66]	[-0.6, 0.6]
Qα	0.7667 (Mixed-1)	0.8963 (Mixed-1)	0.9224 (Mixed-1)
TVC A/B	1.7545 (Mixed-1)	1.7930 (Mixed-1)	1.7962 (Mixed-1)

COMPARISON WITH VEGAMATH

The WCCs obtained using WCAT+VEGACONTROL were evaluated in VEGAMATH by ELV. This allowed assessing the validity of the obtained results as well as serving as way to measure the representativeness of the two simulators. The outcomes of the simulation runs are depicted in Fig. 7, allowing the optimization parameters to vary in the interval [-1, 1].

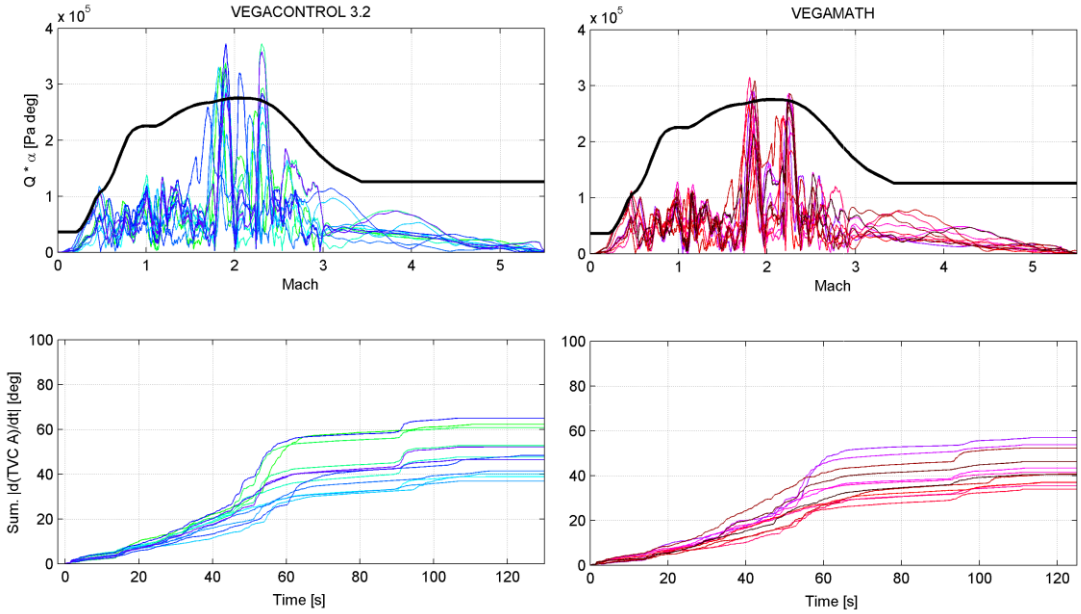

***Fig. 7** Simulation of the Qα WCCs obtained with VEGACONTROL versus VEGAMATH*

Several WCCs violating the Qα assumption were detected in VEGAMATH. In terms of TVC consumption, however, no violations were identified. It is also stressed that maximizing the TVC consumption leads to WCCs that, in some circumstances, violate the Qα criterion. This indicates that the two criteria may be aligned, as expected. Indeed, for conditions in the extremes of the flight envelope, the vehicle is likely to require an increased control effort.

B. MU ANALYSIS SET-UP & RESULTS

The analytical μ-based robustness analyses used as the main analysis model, the symbolic LFT presented in Section IV.C. As aforementioned, this general atmospheric launcher LFT is formed by 6 scatterings 'flags' associated to four real time-varying parameters (*Th*, q_{dyn}, C_N, x_{CP}) and with a total dimension of 31. Based on this main LFT, two other LFT models were derived, each of which with a slightly different analysis purpose. The first LFT, called *perf-LFT*, was oriented towards performance analysis as it considered the actual output vector of the launcher. The other model, called *classtab-LFT*, was used for evaluating classical LTI metrics such as gain/phase margins. The use of these two models allowed building up confidence on the results as well as demonstrating the flexibility of performing different analysis starting with a main symbolic LFT.

In addition, in order to demonstrate the analysis capabilities, advantages and shortcomings of μ-analysis, several types of analyses and uncertainty types were used for each of these two LFT models. Specifically, robust stability (RS) and robust performance (RP) tests were performed using both, real and complex uncertainties. The latter was obtained by adding 1% complex uncertainty to each parameter in the LFT set, which helps improve speed and convergence of the algorithm, as it avoids numerical algorithmic issues with real parameters [1].

For each μ-analysis test, a constant level of uncertainty was applied to all the flags not used in the LFT uncertainty set (this set will be referred to hereafter as *other-flags*) and a time instance in the P80 flight was selected (every second from 1 to 120). This enabled the numerical evaluation of the symbolic LFT and allowed applying the μ-analysis test searching for the worst-case combination of the normalized parameters in the LFT set.

For the first LFT model, the results indicate that the followed analytical worst-case algorithm and process can successfully identify many combinations that result in violation, and even strong violation, of the time-domain Qα-versus-Mach bound. Furthermore, these WCCs are found well spread throughout the range of the parameter LFT set and, indeed, by examining their spread, several violation-regions can be identified. This latter remark is important as it shows that these worst-cases are not improbable since there are regions of violation and thus the probability of encountering them is not negligible. Nonetheless, it is noted that no quantifiable probability can be associated to this conclusion.

With respect to the 2nd LFT model, a comparison between the two models shows that although the first model is better in obtaining strong violations (since it is a more performance-oriented model) the 2nd model obtains many more non-strong violations (since it is a more stability-oriented model). Further, the WC cases

obtained with the two models are different but all show a general trend from which several different WC combinations can be easily identified.

P80 YAW-AXIS PERFORMANCE ANALYSIS

For the *perf-LFT* model, seven campaigns were performed based on the level of uncertainty applied to the *other-flags* set. The first campaign considered the nominal case (*other-flags=0*) and the other campaigns used reduced values of the uncertainty *{-50, -30, -10, 10, 30, 50}%* .

As an example of the results, Figure 8 shows the μ-bounds corresponding to two time instances *{3, 108}* seconds. It is noted that these time instances are just picked for graphical convenience (i.e. ease of visualization of the mu bounds) but are representative of the advantages of μ analysis as both cases are at first impression not the most adequate for worst-case analysis. Indeed, the case at time 3 seconds is still in the vertical flight where velocity is very low and that for 108s is at the end of propulsion in which the thrust is much decreased.

Looking at Figure 9, on the left-column, the real RS and RP results are given, while on the right one, the complex RS are shown (but including the RS-real for comparison purposes). Note that both μ-bounds for the real-RP test at time 3 seconds (top-left plot) reach almost exactly the limiting level (i.e. μ=1) but that at time 108 seconds (bottom-left plot) both upper μ-real RP and RS bounds result in violations around the frequencies $10^{-2.5}$ and $10^{-0.5}$ radians. Since the lower-bounds for the real-RP test reach the μ=1 level for both time instances, it is then possible that when applying the corresponding WCCs to the nonlinear, time-domain simulation a violation will occur. In addition, after further examination, it was observed that if both the upper bounds for the μ-real RS and RP are above or at the μ=1 level, then it is very likely that one of the perturbations will result in a (Qα performance) time-domain violation.

With respect to the complex case, the results indicate that for time 3 seconds the corresponding perturbations (once turned into a real set) should not result in a violation, since all the bounds are below μ=1, while at time 108 seconds all the obtained WCCs could result in performance violations, since they satisfied μ>1.

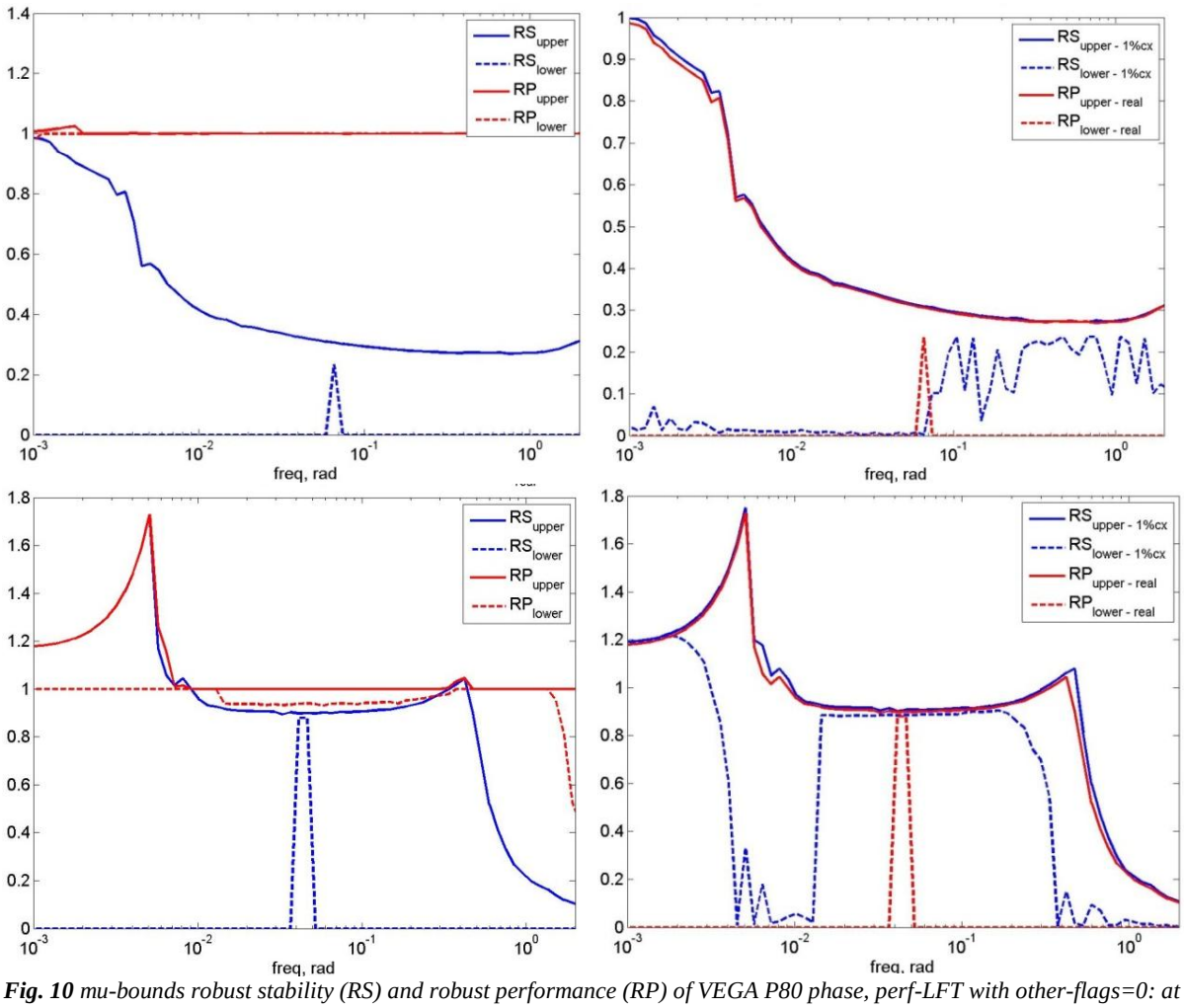

Fig. 10 *mu-bounds robust stability (RS) and robust performance (RP) of VEGA P80 phase, perf-LFT with other-flags=0: at times {3, 108} seconds (top & bottom rows) for real-Δ and 1% complex-Δ (left & right columns)*

Fig. 11 shows the (Qα-vs-Mach) time responses for the WCCs obtained from the μ-analyses shown in the previous figure. Notice that the real-RP perturbation at time 3 and the real-RS at time 108 result in violation of the time-domain bound (solid black line). Moreover, the real-RP WCC at time 108 does not result in a violation nor do the other perturbations at 108 seconds violate the performance limit. It is highlighted that this disparity between the frequency-domain and the time-domain results arises from two sources: (i) the frequency vs. time domain analysis, and more importantly (ii) the time-varying nature of the nonlinear system in the time-domain analysis. Especially for the latter, note that the frequency analysis is performed in a linear-time-invariant frame and at a single time instant (which might be far away or even after the time at which the violation occurs in the time simulation). Nonetheless, it is seen that, despite these differences, violating worst-cases are found using the proposed methodology.

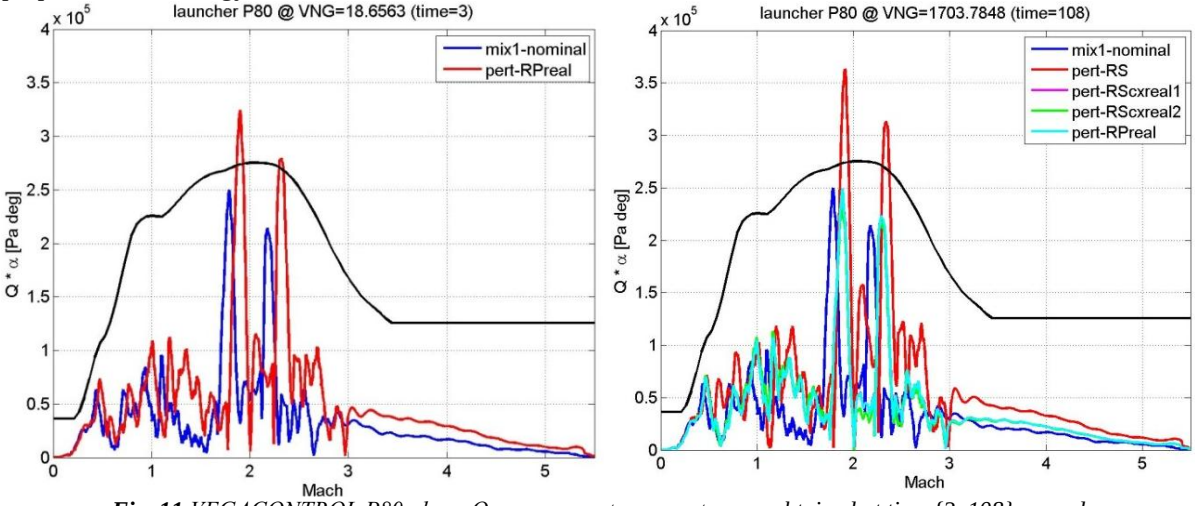

Fig. 11 *VEGACONTROL P80 phase Qα responses to μ worst-cases obtained at time {3, 108} seconds*

P80 YAW-AXIS GAIN/PHASE MARGIN ANALYSIS

In this subsection the LFT-modeling and μ-analysis framework is used to search for WCCs following classical gain and phase margins violations. The process to obtain the *classtab-LFT* model, starting from the general symbolic LFT from Section IV.C, is just a matter of including a multiplicative uncertainty related to an ellipse in the Nyquist plane, which is then multiplied at the input of the P80 yaw-axis model (i.e. the TVC command) [8]. The GM/PM uncertainty is represented by a single additional uncertain parameter in the Δ set together with two additional symbolic constants, *a* and *Wn=-r/a*, placed in the *M* matrix. The symbolic constants are related to the center (*–a*) and radius (*r*) of the Nyquist plane ellipse defined by the GM/PM requirements. Thus, by assigning a numerical value to the required GM and PM, a two-equation system can be solved to yield the corresponding values for *a* and *r* (and thus *Wn*). In the present case, the GM and PM limits for the nominal case (*other-flags=0*) are 3dB and 43 degrees respectively, and arise from VEGA requirements.

Fig. 12 shows the equivalent to Fig. 10 but for the *classtab-LFT* model and only at time instant 3 seconds. It is clearly seen, in comparison to Fig. 10, that, for the 2nd LFT model, the obtained real-RS bounds shoot up to the level μ=1, indicating likely violation in the time-domain of the corresponding WCC.

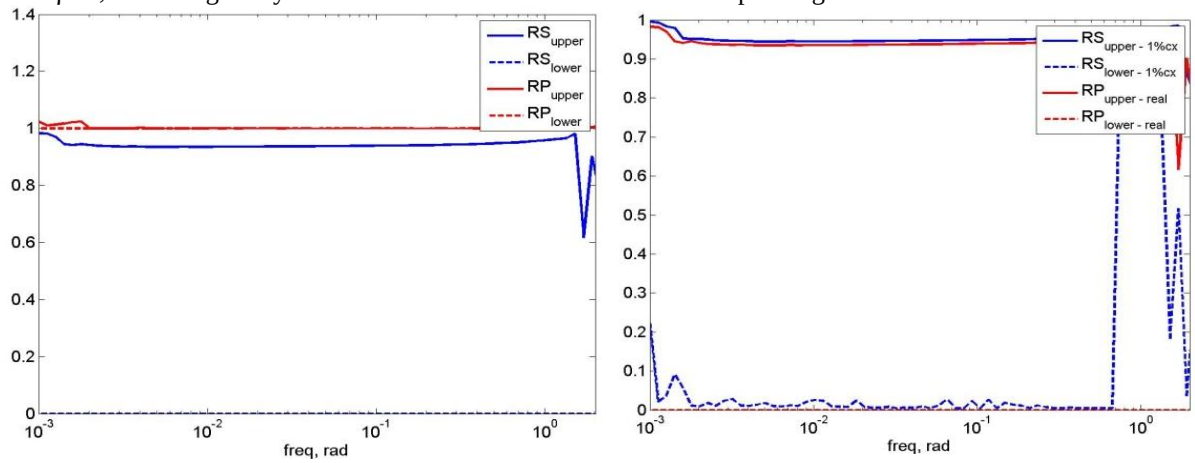

Fig. 12 *mu-bounds RS and RP of VEGA P80 phase, classtab-LFT with other-flags=0 at times 3seconds for real-Δ (left column) and 1% complex-Δ (right column)*

COMPARISON

Fig. 13 shows the VEGACONTROL Qα-vs-Mach "strong" violations (i.e. those $Q\alpha \geq 3$) corresponding to all the WCCs obtained using the *perf-LFT* (left column) and *classtab-LFT* (right column). In grey on the right-upper corner, the time instances at which the WCCs are obtained are listed. It is seen that there are many more from using *perf-LFT* than *classtab-LFT*: 19 instances for the first LFT (at times {*3, 6, 15, 35, 41, 44, 54, 55, 68, 69, 72, 83, 87, 88, 91, 96, 103, 108, 116*} s) versus 8 for the other (at times {*7, 47, 52, 53, 59, 62, 72, 96*} s). The opposite is true for "non-strong" violations ($Q\alpha < 3$), where the *classtab-LFT* yields 58 instances versus 20 for the *perf-LFT*. This has to do with the fact that for the former we are seeking to maximize the stability violation which in the true nonlinear, time-varying case might not have sufficient time to develop resulting in a "non-strong" violation that only at times is "strong". Similar results were obtained by ELV using VEGAMATH.

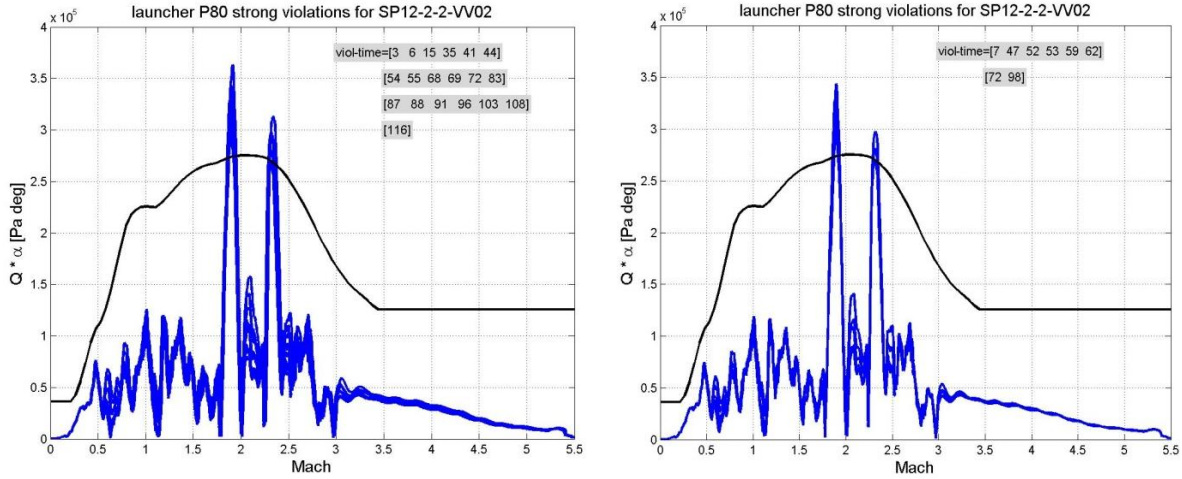

Fig. 13 *VEGACONTROL P80 phase Q*α responses to μ worst-cases from perf-LFT (left) and classtab-LFT (right)*

VI. CONCLUSIONS

This paper described the methods adopted in enhancing the state-of-practice for launcher V&V with a focus on those methods used for the detection of worst-case conditions that may significantly degrade the performance of the VEGA launch vehicle and, ultimately, lead to instability. One of the approaches selected considers the optimization of certain costs functions that are intrinsically aligned with the performance of the vehicle, while the other adopted linear analysis tools based on the structured singular value formalism. The optimization campaigns were set up based on the WCAT toolbox (using the option of Hybrid Differential Evolution genetic algorithms), but preceded by an approximate sensitivity analysis of each criterion to all and each of the variables.

Several worst-cases were obtained, with both approaches, which violated the Qα criteria. In particular, the bending modes present in VEGACONTROL seem to increase the number of WCCs violating the requirements of the launch vehicle. Moreover, WCCs violating the Qα criterion were also captured even if the variables are constrained to the interval [-0.6, 0.6] showing a certain lack of robustness.

From a physical interpretation, the worst-case conditions obtained are plausible, in the sense that some of the parameters optimized are typically selected to detect WCCs. Indeed, the dispersion and uncertainty for the axial coefficient ('disp_CA' and 'unc_CA') are set to -1, which implies diminished drag and, thus, an increase in airspeed and dynamic pressure. Moreover, the +1 scattering on 'air_density_scat' means that the air density is at its upper limit, which consequently leads to increased dynamic pressure (and thus, more likely violation of the Qα criteria).

The impact of the wind sequence on the overall results indicates that jointly searching for the worst wind profile and the parameters combination, may yield worst-case conditions that are more severe than the ones detected in this project.

Simulating the WCCs in VEGAMATH generates results that are similar to those of VEGACONTROL, which render the latter simulator a useful tool in the controller design and V&V process of the VEGA launcher.

ACKNOWLEDGEMENTS

This work is part of a European Space Agency (ESA) study led by Deimos, and participated by ELV and DLR, entitled “Robust Flight Control System Design Verification and Validation Framework (RFCS)”. This study was established with the objective of developing, demonstrating and comparing with a traditional V&V framework a new enhanced design V&V framework through their application to the V&V of a complex launch vehicle, specifically to the VEGA launcher.
This work was funded by the European Space Agency under ESA-ESTEC contract 1-6322/09/NL/JK.

REFERENCES

[1] Balas, G.J., Doyle, J.C., Glover, K., Packard, A., Smith, R., "μ-Analysis and Synthesis Toolbox," Musyn-Mathworks, 1998
[2] Bateman, A., Ward, D., Balas, G., "Robust/Worst-Case Analysis and Simulation Tools," AIAA GNC, San Francisco, 2005
[3] Belcastro, C. and Belcastro, C., “On the Validation of Safety Critical Aircraft Systems, Part II: An Overview of Analytic and Simulation Methods,” AIAA GNC Conference, August, 2003
[4] Doyle, J., Packard, A., Zhou, K., “Review of LFTs, LMIs, and μ,” IEEE Conference on Decision and Control, 1991
[5] Fielding et al., "Advanced Techniques for Clearance of Flight Control Laws," Lecture Notes in Control and Information Science. Berlin: Springer-Verlag, 2002
[6] Hanson, J.M., Beard, B.B., "Applying Monte Carlo simulation to launch vehicle design and requirements verification," Journal of Spacecraft and Rockets, 49(1), 136-144, 2012
[7] Hansson, J., “Using Linear Fractional Transformations for Clearance of Flight Control Laws,” MS Thesis Linköpings University (Sweden), LiTH-ISY-EX-3420-2003, October 2003.
[8] Jacklin, S.A., Schumann, J.M., Gupta, P.P., Richard, R., Guenther, K., and Soares, F., “Development of Advanced Verification and Validation Procedures and Tools for the Certification of Learning Systems in Aerospace Applications,” Infotech Aerospace Conference, Arlington, USA, 2005
[9] Lambrechts, P., Terlouw, J., Bennani, S., Steinbuch, M., “Parametric Uncertainty Modeling using LFTs,” American Control Conference, San Francisco, USA, 1993
[10] Magni, J.F., "Linear Fractional Representation Toolbox: Modeling, order Reduction, Gain Scheduling," TR 6/08162 DCSD, ONERA, Toulouse, France, January, 2004
[11] Marcos, A., Balas, G.J., “Development of Linear Parameter Varying Models for Aircraft,” AIAA Journal of Guidance, Control, and Dynamics, vol. 27, no. 2, pp. 218-228, 2004
[12] Marcos, A., Bates, D.G., Postlethwaite, I., “A Symbolic Matrix Decomposition Algorithm for Reduced Order Linear Fractional Transformation Modelling”, Automatica, vol. 43, no. 7, pp. 1125-1306, 2007
[13] Marcos, A., García, H., Mantini, V., Roux C., Benanni, S., “Optimization-based worst-case analysis of a launcher during the atmospheric ascent phase,” AIAA GNC, 2013
[14] Marcos, A., Mantini, V., Roux, C., Bennani, S., "Bridging the Gap between Linear and Nonlinear Worst-Case Analysis: An Application Case to the Atmospheric Phase of the VEGA Launcher", IFAC ACA, 2013
[15] Menon, P.P., Prempain, E., Postlethwaite, I., Bates, D.G., “Nonlinear Worst-Case Analysis of an LPV Controller for Approach-Phase of a Re-Entry Vehicle,” AIAA GNC, 2009
[16] Storn R., Price, K., “Differential evolution: a simple and efficient heuristic for global optimization over continuous space”, Journal of Global Optimization, Vol.11, pp. 341–369, 1997