M. Ganet-Schoeller
(Ariane Group)

J. Desmariaux
(CNES)

C. Combier
(ISAE)

E-mail: martine.ganet@airbusafran-launchers.com

DOI: 10.12762/2017.AL13-08

Structured Control for Future European Launchers

Control of a flexible launcher during the atmospheric flight phase is a highly challenging control problem involving multiple and concurrent design requirements: stability (stabilization of unstable rigid dynamics, sloshing modes and flexible structural modes), performance (guidance tracking, structural load minimization) and robustness (physical parameter uncertainties and accommodation to multiple vehicle configurations) on a non-stationary system. This paper focuses on co-funded AG/CNES research activities on the development of an advanced modular control strategy using recent advances in structured control design. We demonstrate here that structured H_∞ synthesis could give a gain scheduling solution to full time-varying flexible launcher control problems during the non-stationary ascent phase, enabling load performance improvement between complex sets of requirements, and, design cost improvements through simplification of the tuning process. We also present a generic framework for rapid control design that is now applicable both for European launcher families already in activity (with existing S/W and fixed controller structure) and for future expendable and reusable launchers.

Introduction

Ariane Group (AG) and the CNES (French Space Agency) aim to safeguard the exceptional quality and reliability of Ariane 5, while developing a family of next-generation rocket launchers designed to consolidate Europe's leadership in the space industry. In this frame, this paper focuses on co-funded AG/CNES R&D activities for developing *launcher advanced control strategies* applicable both for launchers already in activity (with existing S/W and fixed controller structure) and for future expendable and reusable launchers. Motivation for this research is twofold: improve performance and reduce development cost focusing on modularity and accommodation to multiple vehicle configurations.

Conventional launcher control design during the ascent phase is a challenging robust control problem. Key challenges to be tackled to ensure performance and robustness are:

- aerodynamic instability during atmospheric flight, which represents a high level of risk [1],
- presence of multiple badly damped bending and sloshing modes,
- disturbances from the external environment (mainly wind turbulences),
- non-linearity of sensors and actuators (delays, noises, limited control authority, etc.),
- uncertainties and dispersions on all physical parameters that characterize launcher dynamics, actuators and sensors,
- mission dependency of all the parameters; e.g., launcher characteristics such as MCI and bending modes depend on the payload,
- varying launcher characteristics throughout the flight (mass, thrust, aerodynamics, bending modes, etc.), as well as varying objectives (perturbations attenuation, consumption, accuracy, etc.) and constraints (loads, actuator limitations).

Therefore, to improve launcher performance, a first research direction is to improve model knowledge and accuracy, which is usually done on operational launchers using post flight analysis as per ARIANE or VEGA. Another research direction is to use recent developments in robust control methods. Indeed, during the two last decades, Europe has been successfully working on robust control techniques, such as Ariane 5 LQG and H_∞ control, the VEGA robust modal control, adaptive control, optimization methods, or LPV control. Research and applications were also performed all over the world. Complementary results on adaptive control and optimization methods for flexible launcher control of the Ares launch vehicle could be found in the literature. All of these design methods were successfully tested and presented at various AIAA, IFAC, ESA or EUCASS conferences. However, despite their satisfying performances, their implementation could be complex due to the high order of the synthesized controllers required to ensure the desired dynamics and performance.

It was not until recently that structured control developments opened new perspectives for control design: by combining *robust control with controller structure requirements*, it is now possible to directly synthesize a low-order controller or fixed-structure controller. This problem is often that of a non-convex and typically non-smooth (non-differentiable) optimization [2, 3]. Recent research has led to the development of new powerful tools, such as structured synthesis `hinfstruct` *[3]* and `systune` [4,10], available in the Matlab Robust Control Toolbox (RCT) [15].

The purposes of this paper are, on the one hand, to demonstrate that structured H_∞ synthesis could solve full time-varying flexible launcher control problems, enabling performance and cost improvement and, on the other hand, to present the resulting generic framework for rapid launcher control design. This framework was developed on a representative benchmark before being used both for the improvement of existing launcher control laws and for future launcher pre-development phases.

This paper is organized as follows: in Section 3 we give an overview of the launcher control problem, then in Section 4 H_∞ and structured H_∞ theory are recalled, focusing on their respective advantages and drawbacks for launcher control. A generic framework for rapid launcher control design is presented in Section 5. Sections 6 and 7 gather results from various applications.

Launcher control

Launcher control overview

The automatic control of a launcher is one of the four main functions of the overall flight-control system, which also encompasses redundancy, navigation and guidance functions. Since the roll, pitch and yaw axes are weakly coupled, the control design is based on the assumption that each axis can be controlled independently of the other two. *A-posteriori* verifications of global performance and stability are, of course, performed. An example of a 1-axis control loop for atmospheric flight is depicted for illustration in Figure 1.

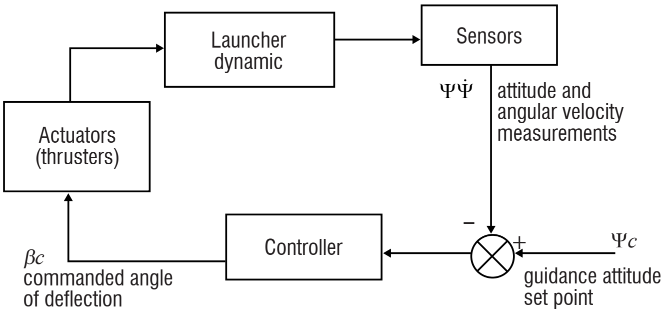
Launcher dynamic
Sensors
Actuators (thrusters)
$\Psi\dot{\Psi}$
attitude and angular velocity measurements
Controller
–
+
Ψc
βc
commanded angle of deflection
guidance attitude set point

Figure 1 – Flight control loop (one axis)

In this figure, corresponding to the launcher benchmark, inputs for the controller are the angular and angular rate measurements (additional acceleration measurement could also be used as proposed in [5]). The controller outputs a commanded thrust deflection angle. As far as the control function is concerned, the atmospheric phase (flight from lift-off to the jettisoning of the solid propellant boosters) is the most critical one. During this phase, the control requirements are, by decreasing order of importance:

- to ensure the stability of the launcher rigid, bending and propellant sloshing modes, with sufficient stability margins,
- to compensate for external (wind, wind gusts) and internal (thrust misalignment, static error of the servo-actuators, thrust asymmetry) disturbances, while minimizing angle-of-attack, for structural sizing reasons,
- to follow the guidance orders (attitude set points), by ensuring a static error and a response time compatible with guidance requirements,
- to minimize the cumulated commanded thrust deflections (hereafter called consumption), since the hydraulic activation devices use a blow-down system.

All of the uncertainties and dispersions associated with launcher and trajectory parameters (mass, inertia, location of the center of gravity, bending and sloshing modes, propulsion and aerodynamic characteristics, etc.) must be taken into account for the tuning of the control law. All of these parameters also fluctuate during the flight, which makes the control problem essentially non-stationary (time-variant).

The control function must therefore simultaneously fulfill:

- *Performance requirements:*
  the control law must ensure the required launcher flight behavior, while compensating for the various disturbances.

- *Robustness requirements:*
  this behavior must be preserved regardless of the internal and external fluctuations that may adversely affect the vehicle during the flight.

The main issues of launcher control lies in the trade-off between these two sets of requirements that are essentially contradictory, since an ill-known system cannot be controlled in a highly efficient way.

Launcher applications

The generic framework for rapid launcher control design was developed on a representative benchmark before being applied to the European launcher control design. This benchmark that can now be used, on request, for internal and external research studies on control design and validation, is presented here. European launcher models and data are not presented here for industrial confidentiality reasons; however, for the development of the generic framework, their main added value concerns the fixed discrete time controller structure, as well as the number and type of actuators and sensors.

Launcher benchmark models

This benchmark deals with pitch control of a symmetric launcher during the atmospheric flight phase from take-off to tail-off. Launcher dynamic equations are linearized around reference trajectory in the body frame

(Figure 2), leading to a state space representation in continuous time defined by Equation (1). Launcher benchmark dynamics contain:

- rigid mode dynamics defined by a bi-dimensional linear perturbation model already described in detail in the literature [6],
- bending modes represented by a second-order model with low damping,
- an actuator model described by a second-order transfer function,
- sensor delays (IMU and gyrometer) that are directly included during discretization of continuous time launcher models.

$$\begin{cases} \dfrac{d}{dt}X = AX + BU \\ Y = CX \end{cases} \tag{1}$$

With

$$X = \begin{pmatrix} \Delta\dot{\theta} \\ \Delta\theta \\ \Delta\dot{z} \\ \Delta\dot{\beta}_R \\ \Delta\beta_R \\ \dot{q}_i \\ q_i \end{pmatrix} \quad U = \begin{pmatrix} \Delta\beta_c \\ \Delta W \end{pmatrix} \quad Y = \begin{bmatrix} \Delta\theta_m \\ \Delta\dot{\theta}_m \end{bmatrix}$$

$$A = \begin{pmatrix} 0 & A_6 & A_3 & 0 & K_1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 & A_2 & 0 & K_2 & 0 & 0 \\ 0 & 0 & 0 & -2\xi_\beta\omega_\beta & -\omega_\beta^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\xi_\beta\omega_\beta Mb_i & -\omega_\beta^2 Mb_i - P_c htu_i & -2\xi_i\omega_i & -\omega_i^2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & B_1 \\ 0 & 0 \\ 0 & B_2 \\ \omega_\beta^2 & 0 \\ 0 & 0 \\ \omega_\beta^2 Mb_i & 0 \\ 0 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & -hp_{IMUi} \\ 1 & 0 & 0 & 0 & 0 & -hp_{GYi} & 0 \end{pmatrix}$$

$$\text{Where} \begin{cases} A_1 = -\dfrac{P_C}{m} + (T - D) \\ A_2 = -\dfrac{D}{V_R} \\ A_3 = \dfrac{A_6}{V_R} \\ A_6 = \dfrac{mDL_F}{I} \end{cases} \quad \text{and} \begin{cases} K_1 = \dfrac{P_C l_{tu}}{I} \\ K_2 = -\dfrac{P_C}{m} \\ B_1 = -\dfrac{A_6}{V_R} \\ B_2 = -A_2 \end{cases}$$

All rigid and bending mode coefficients of Equation (1) are *time-varying* along a given trajectory; they are also subject to *uncertainties* leading to more than 20 uncertain parameters. Finally, *three payload* configurations are considered (1000 kg, 2000 kg and 6000 kg).), with their associated impact on rigid and flexible mode characteristics. For each payload, a *Model DataBase (MDB)* covering the worst uncertain cases is included in the benchmark.

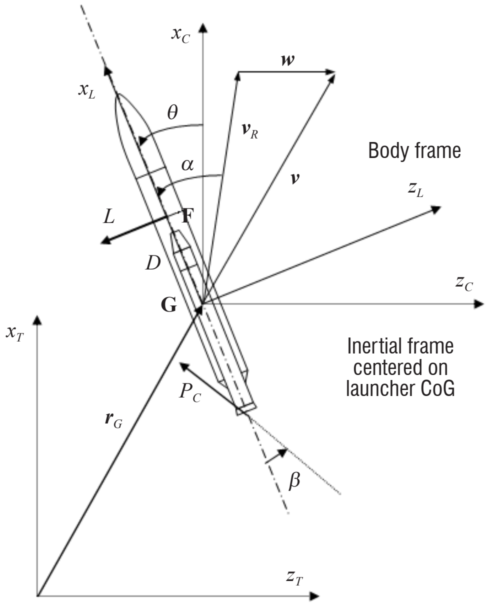

Figure 2. Launcher reference frames

Launcher control objectives and validation means

Stability requirements

Stability margins shall be computed using the Nichols or Bode diagrams *in discrete time.* The stability margin requirements shall be respected for the entire set of pre-defined *worst cases* located on the bounds of the uncertainty domain, as described in the user manual, for all instants and for all payload configurations.

	Stability margins
LF open-loop gain margin	1 dB
HF open-loop gain margin	3 dB
Phase margin	-
Delay margin	50 ms
Bending mode delay margin	50 ms
Bending mode gain margin	Overshoot < -6 dB

Table 1 – Stability margin requirements Performance requirements

The control function shall guarantee the following stability margins (with equipment characteristics) on the SISO system.

Performance requirements

All of the *time domain requirements* shall be respected for the time varying launcher, in the *nominal case*, for the whole set of payload configuration and for 4 different wind profiles. This validation shall be done using the Simulink models delivered with the benchmark.

Disturbance rejection

Disturbance rejection need is mainly linked to atmospheric wind disturbance impact on angle of attack, and to the measurement noise impact on cumulated deflection.

- The control function shall maintain the induced aerodynamic angle of attack compatible with general load specification $Q\alpha < 150$ kPadeg.

- Thruster misalignment impact on load shall not exceed 10% of the specification.
- Cumulated deflection shall remain lower than 200° during the 100s of the flight phase.

Guidance tracking constraints

In the steady state, under nominal conditions, the control function shall be able to control the attitude with the following accuracy:

- attitude $\leq$ 2° (compromise with $Q\alpha$ minimization),
- attitude rate $\leq$ 1°/s during thrust steady state and 0.5°/s at the end of tail off.

Technical constraints

The controller shall be implemented in discrete time with a sampling frequency of 20 Hz.

Actuator constraints:

- maximal deflection angle $\leq$ 6°,
- maximal deflection rate $\leq$ 15°/s.

Reference H_∞ controller

In this benchmark, we have defined a non-structured gain scheduled H_∞ control law that will serve as a reference. This reference controller was fully validated and respects all stability and performance objectives; even if a small margin is left for optimization.

H_∞ and structured H_∞ synthesis – theory and analysis

In this section, we give an overview of (non-structured) H_∞ theory and structured H_∞ theory focusing on their respective advantages and drawbacks for launcher control application.

H_∞ synthesis

H_∞ synthesis was developed in the eighties [7]. It is based on the use of the H_∞ norm, which measures the maximum amplification that a system can apply to any input signal. For SISO systems, this norm is equivalent to the peak gain value. H_∞ synthesis is a frequency domain robust control method applicable for MIMO, LTI, causal and proper systems, that guarantees nominal stability, nominal performance and robust stability. H_∞ synthesis uses the standard form of Figure 3.

Where $P(s)$ is the system model augmented by objective-linked weighting functions, $C(s)$ is the controller; u is the command vector, y is the measurement vector, $w=[w_1,w_2,...,w_N]^T$ is the disturbance vector and $z=[z_1,z_2,...,z_M]^T$ is the error vector that will be minimized.

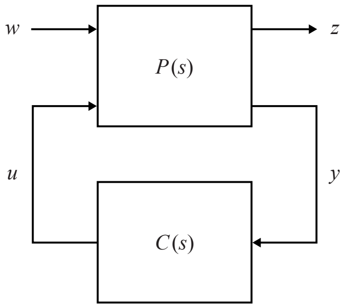

Figure 3 – H_∞ standard form

H_∞ synthesis sub-optimal problem is to find a controller C (or K) that internally stabilizes the augmented system P and such that:

$$\left\| \frac{z}{w} \right\|_\infty < \gamma \tag{2}$$

Many solutions exist for this synthesis, which can be transformed into a convex problem by means of additional variables, through γ-iteration in continuous time [7] or in discrete time [8], or, LMI approaches [9]

Numerous examples using H_∞ controllers are described in the literature and those controllers are currently used in industry. With regard to launcher control, H_∞ synthesis was successfully developed and implemented on Ariane 5 following a test flight in 2001. During this development, performance gain and development cost reduction were demonstrated through H_∞ synthesis roll-off effect and frequency domain compromise between objectives; however, some drawbacks were identified:

- The H_∞ controller order is equal to the augmented system order; therefore, it directly increases with the level of representativeness of the model and with the number of objectives. Thus, for the Ariane 5 application, design choices were limited to avoid controller post-reduction.

- Minimizing the whole transfer matrix including non-diagonal terms could induce conservatism and tuning difficulties when considering multiple objectives, which was the case for Ariane 5 control.

Non-smooth optimization and structured H_∞ synthesis

Structured H_∞ synthesis uses non-smooth optimization techniques to locally solve H_∞ synthesis problems under additional structural constraints on the controller. Structured H_∞ uses the standard form described in Figure 4. This form is similar to the non-structured H_∞ form, except that the transfer functions are decoupled.

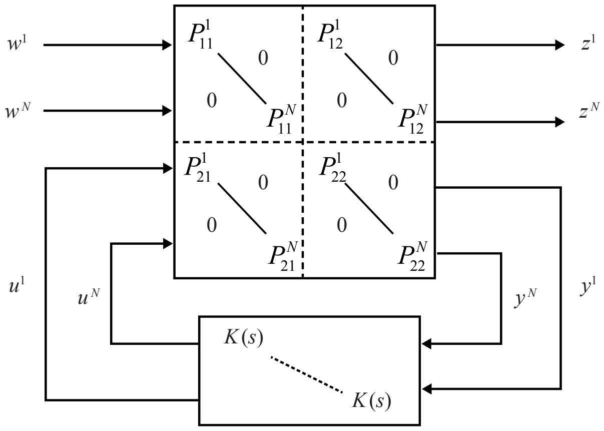

Figure 4 – Structured H_∞ standard form

Structured H_∞ synthesis consists in finding an internally stabilizing controller such that:

$$\exists \ K : \min \left\| \begin{bmatrix} F_l(P,K)_{11} & 0 & \dots & 0 \\ 0 & F_l(P,K)_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & F_l(P,K)_{NN} \end{bmatrix} \right\|_\infty \qquad (3)$$

$$\Leftrightarrow \min_{x \in \Re^k} \max \left(\left\| F_l(P,K)_{11} \right\|_\infty , \left\| F_l(P,K)_{22} \right\|_\infty , \dots , \left\| F_l(P,K)_{NN} \right\|_\infty \right)$$

Where K is the structured controller and the vector x contains all tunable elements of K.

The problem described above is usually that of a non-convex and non-differentiable optimization. It was not until recently that algorithms and solvers for this type of problem began to appear [2, 3]. To date, as far as we know, there are two MATLAB® packages that are capable of addressing this problem via non-smooth optimization: the *HIFOO* packages [2] and the RCT packages `hinfstruct` *[3] and* `systune` [4,10]. In this paper, we will expand work processed with `hinfstruct` in [11] and tackle the launcher control problem with `systune` [4].

Structured H_∞ Advantages and Drawbacks for launcher control

For launcher applications, fixed-order controllers using non-smooth H_∞ algorithms solve most of the drawbacks of H_∞ synthesis. Indeed, its main advantages are:

- use of a *reduced-order fixed-structure* control law,
- *direct quantification* of stability and performance requirements in simple weighting functions, thanks to multi-model approaches and frequency domain limitations, with loop shaping and/or sensitivity function criteria,
- use of Soft/Hard constraints, enabling the automatic minimization of structural loads and consumption.

The disadvantages of fixed-order H_∞ synthesis methods for launcher applications are fading with recent developments and applications.

- One remaining drawback for non-stationary launcher control design is that, today, to our knowledge, *no LPV design* for a fixed-structure controller is sufficiently mature. An alternative is the use of gain surfaces, developed and applied in [12], or, applied in [13]; however, this approach could lead both to an increasing number of controller parameters and to some conservatism linked to gain surface selection. Additional developments in this field would be of great interest. However, in practice, we will show in this paper that, with an adequate initialization process and additional constraints on the controller pole and zero characteristics, the *gain scheduling* approach traditionally used for non-structured H_∞ launcher control is also well suited for structured control design.

- *Local optimization:* non-smooth optimization algorithms are only local algorithms and there is no guarantee of convergence towards a global optimum. *Non-repeatability* can be an issue for industrial implementations of gain scheduling control. This drawback was softened by considering additional constraints for gain scheduling control design. This last point is the only one that still requires attention from an industrial point of view.

Control design framework

The generic framework for control design was developed on the launcher benchmark; it contains both specific functions depending on launcher application, with dedicated interfaces with launcher data, modeling and validation tools, and generic functions for controller design unrelated to launcher applications. Its architecture is described in Figure 5, and the main functions are detailed in this section.

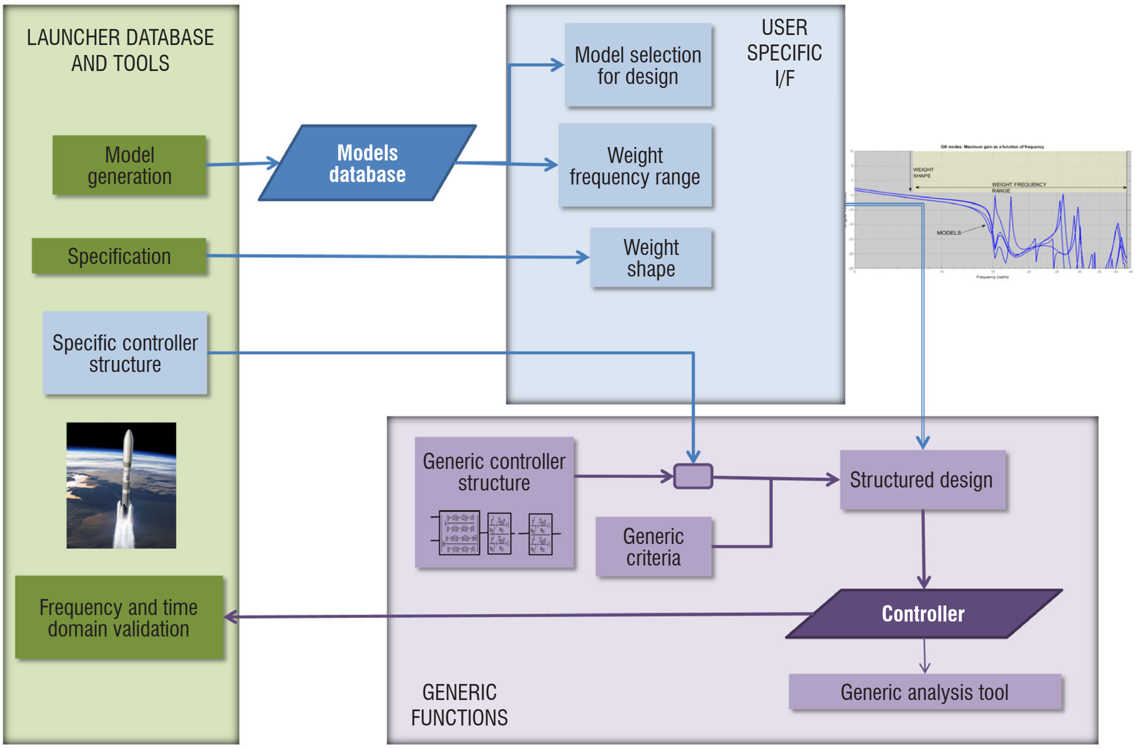

Figure 5 – Control Design Framework

Model generation and selection

A complete launcher Model DataBase (MDB) is generated covering rigid and flexible mode dynamics, nominal and uncertain cases, in continuous time and/or discrete time representations. Then, for each requirement, a single model or multiple models are selected for control design, e.g., worst uncertain case for the Low Frequency (LF) stability margin requirement, worst uncertain case for High Frequency (HF) stability margin requirement, etc.). These models also serve for frequency range definition of weighting functions, as described in the objective transcription.

Controller structure

The controller structure can be specific, or the user can select the default structure defined in [11] for the launcher benchmark. In this case, a MISO structured controller is selected that makes use of both the attitude and angular rate measurements to compute the commanded thruster deflection. Controller structure is separated into a so-called "rigid mode controller" for low-frequency rigid-dynamics control and a "bending mode filter" for high-frequency command filtering.

The generic controller structure contains:

- for rigid mode control, a reduced order controller with constraints on the pole and zero characteristics to avoid interpolation issues,
- for bending mode filtering, a reduced order filter, for which the range of variation of the poles and zeros is limited to enable gain scheduling and to avoid filtering overshoots.

Objective transcription

Problem definition and tuning of robust flexible launcher control design are greatly simplified thanks to the use of structured design: each objective is associated with a generic criterion (transfer function independent from the launcher model) and with mission data (models, frequency range or numerical objective value).

Structured H_∞ design generic criteria

All of the control stability, robustness and performance objectives were translated into *generic criteria* within the `systune` [4, 10] function framework. In this approach, both traditional H_∞ techniques and loop shaping were used, combined with multi-model design and frequency-limited requirements. The traditional approach [14] uses a closed loop sensitivity function S, KS, KG or T described in Figure 6, while loop shaping uses an open-loop transfer function K or KG. Generic criteria used for launcher control design are gathered in Table 1.

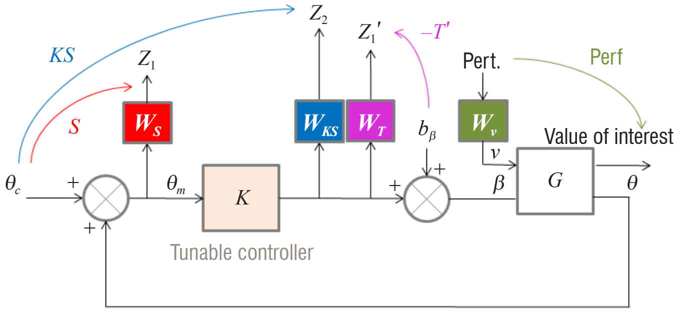

Figure 6 – Closed loop sensitivity functions

Mission data

For each of these generic criteria, a user I/F enables the selection of the following:

- worst case models in launcher MDB,
- the frequency range, determined by model analysis; e.g., the bending mode control objective is applicable for the bending mode frequency range over the launcher configurations,
- once the frequency range has been selected, launcher control requirements (or mission specification) are directly used for weighting function shape definition.

For each objective, Table 2 gathers the constrained sensitivity function used in the classical robust control approach and models that could be used for design.

Objective	Models (MDB)	Sensitivity function or transfer function
LF Margin	Worst Case 1	*S*
HF Margin	Worst Case 2	*S*
Bending mode passive stabilization (gain control)	N uncertain models / Mission-dependent	*KG*
Bending mode active stabilization (phase control)	N uncertain models / Mission-dependent	*T*
Consumption and filtering authority	NA	*K*

Table 2 – Structured H_∞ criterion, models and sensitivity functions

Controller synthesis

Depending on the sampling frequency, it could be advisable to perform the launcher control design in discrete time. Therefore, the use of both continuous and discrete time versions of the `systune` [4, 10] function was validated during the development of this framework.

Gain scheduling was applied using the following process:

- Structured control design for one reference flight point (instant of maximum dynamic pressure).
- We use an automatic load minimization criterion with guarantee of compliance of all other requirements (directly for stability margins and frequency domain requirement / indirectly for time domain requirements); therefore, *reducing the cost of iterative design.*
- Ascent front synthesis with reasonable time interval from the reference flight point until the end of the flight, with initialization using the controller from the previous instant.
- Descent front synthesis with reasonable time interval from the reference flight point down to the beginning of the flight, with initialization using the controller from the next instant.

Then, the set of structured controllers are linearly interpolated along the flight.

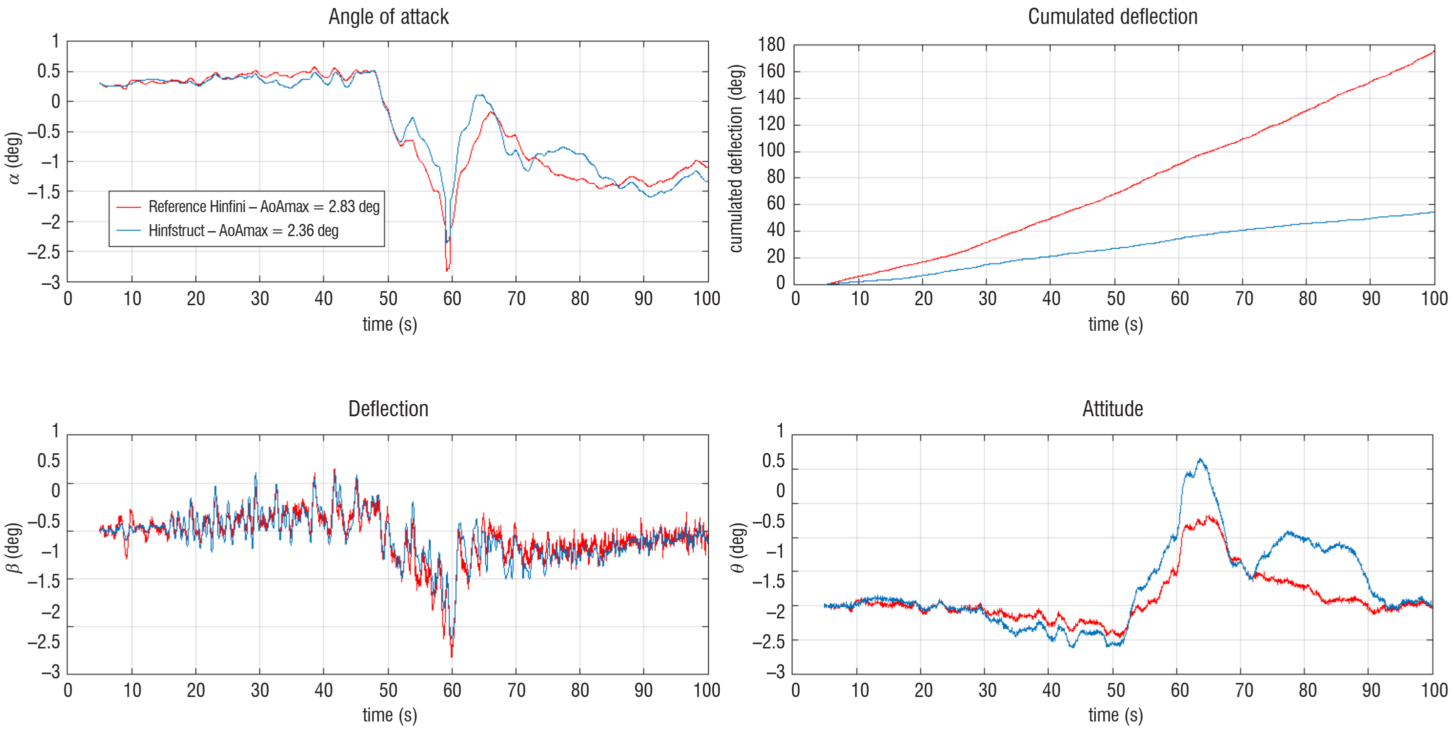

Figure 7 – Simulation results for the reference controller and gain-scheduled structured controller

We have included slight evolutions in the weighting functions and requirements along the trajectory, in order to take into account the objective variation during the flight.

Analysis

A set of generic analysis functions are associated with control design tools for fast validation on control design models. Complete validation is then performed on full MDB with industrial validation tools.

Feasibility demonstration – benchmark application

In this section, structured H_∞ control design feasibility is demonstrated on the launcher benchmark. A summary of structured H_∞ results obtained for the entire flight phase and for the three payload cases is presented and compared with the reference non-structured H_∞ controller. These results were extended to other applications in Section 7, thereby validating the genericity of the control design framework on different launcher configurations. These results also illustrate the advantages of structured design for rapid controllability and control design.

One can see, in the simulation results of Figure 7 that the gain scheduling process, using gain scheduled weighting functions, is quite efficient, without presenting interpolation issues.

All stability requirements, both for rigid dynamics and bending modes, are respected (Figure 8), as well as all performance requirements (Figure 9). In this last figure, performance of the reference non-structured controller is also plotted, highlighting loads and consumption improvements. Indeed, the gain-scheduled structured controller design leads to a reduction in the loads of about 5 to 15%, depending on wind disturbance, and a consumption reduction of about 60% with respect to the reference controller.

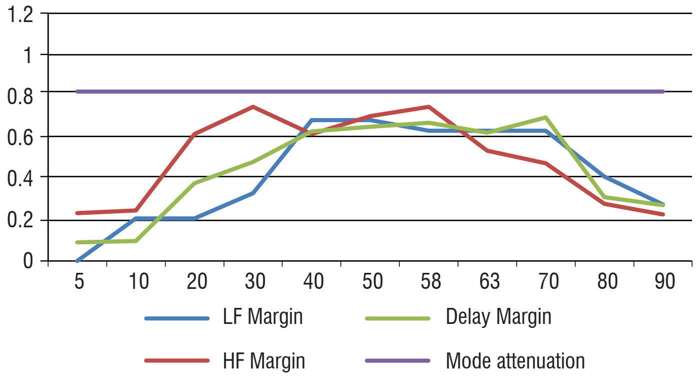

Figure 8 – Gain-scheduled structured controller - Normalized stability margins (with respect to the requirement)

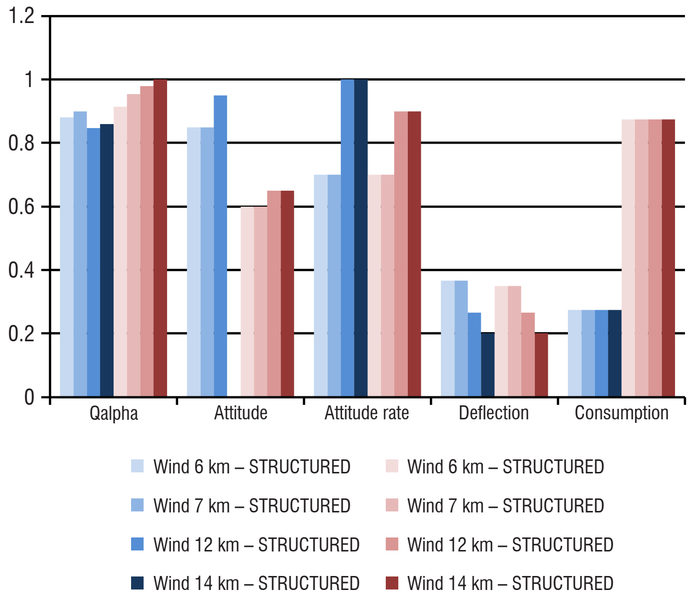

Figure 9 – Gain-scheduled structured controller and reference controller - Normalized performance (with respect to the requirement)

European launcher applications

In this section, we present some results of this generic framework, obtained from a multipurpose application for different launcher configurations; these results illustrate its strengths for rapid control design.

On the one hand, simplification in the control design process that was shown during the development of the launcher benchmark was confirmed on different European launcher configurations. Low and high frequency control design requirements are simultaneously tackled on worst cases extracted from each launcher MDB without design iterations. Resulting performances are always better, or at least equivalent to those obtained with traditional approaches. Figure 10 illustrates the time-varying compromise between low and high frequency objectives on the Bode plot of the controller for an IMU measurement. Figure 11 clearly shows the respect of stability margin requirements for all of the instants and models used for the design. Figure 12 demonstrates the respect of load requirements for a complete real wind database measured at the Kourou launchpad.

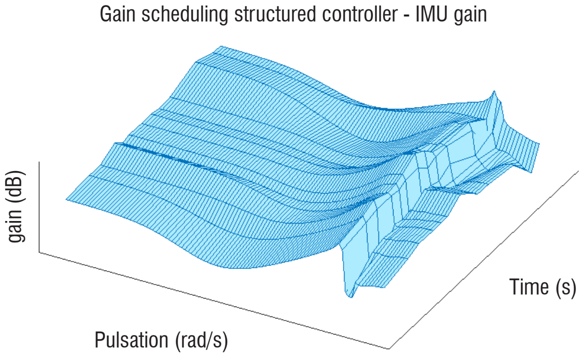

Figure 10 – Example of IMU controller Bode plot

On the other hand, these tools are perfectly suited to perform numerous trade-offs and concept choices for future launchers through preliminary performance analysis in an automated way without manual retuning. They were used for:

- automatic trade-off between sensor selection and location, with performance assessment, thereby improving the process proposed in [5],
- trade-off between robustness and performance requirements. An example of the impact of stability margin relaxation on angle of attack performance is shown in Figure 13,
- trade-off between phase or gain control of the first bending mode. An example of the impact of gain or phase control of the first bending mode on the angle attack performance is shown in Figure 14,
- trade-off between mission dependence and robustness.

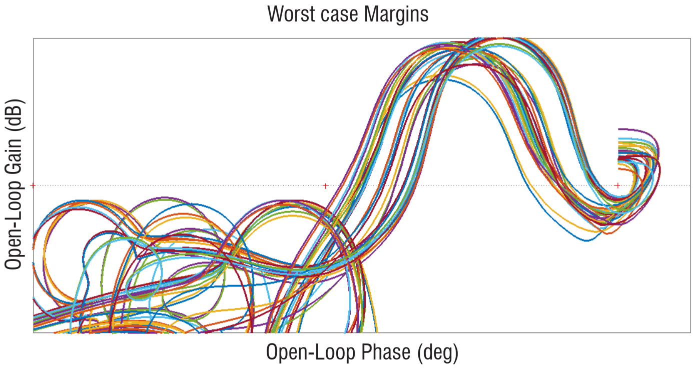

Figure 11 – Illustration of stability margin requirement respect

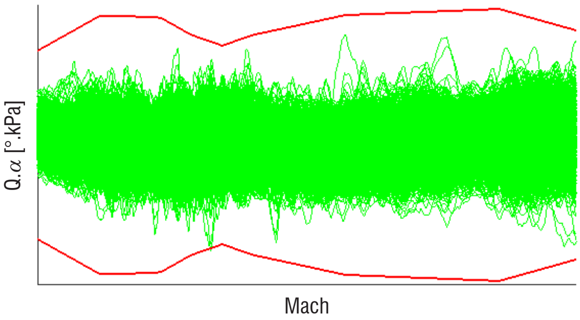

Figure 12 – Illustration of load requirement respect when facing real winds at Kourou

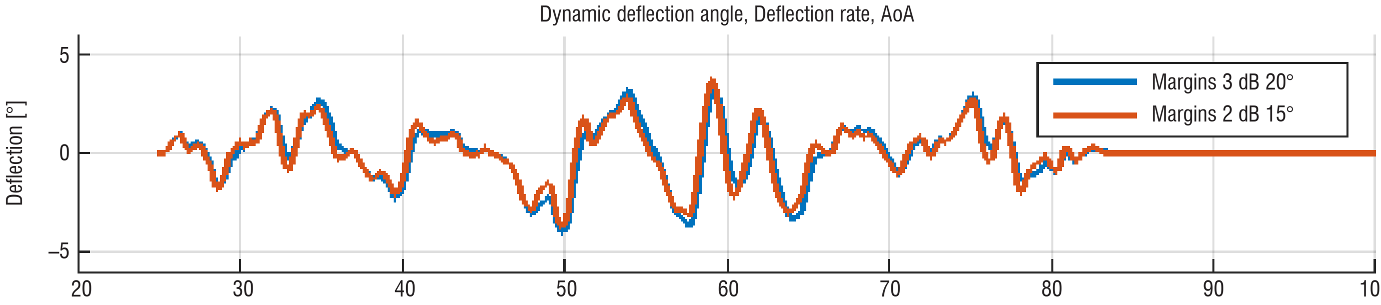

Figure 13 – Trade-off between stability margins and performance results

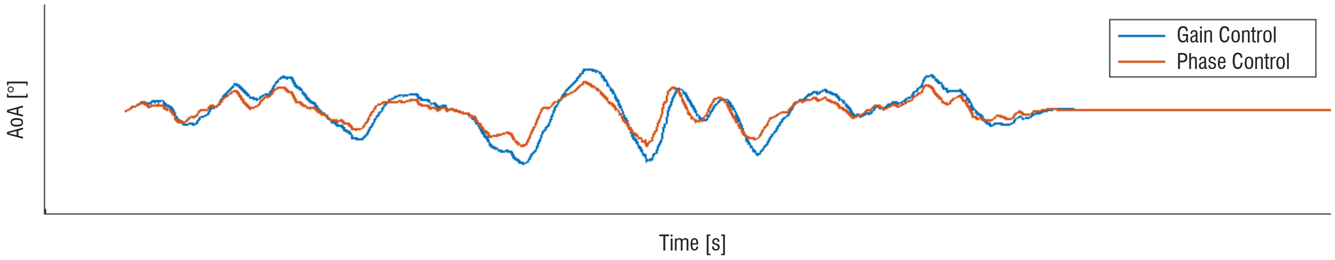

Figure 14 – Trade-off between bending mode gain or phase control and performance results

Conclusion

In this study, we have taken advantage of all of the enhancements of structured control design (fixed controller structure, multi-model, band-limited objectives, Soft/Hard constraints, etc.) to develop a *generic framework for rapid control design*. This framework was developed on a representative benchmark (with complete freedom on controller structure) before being proposed for the improvement of existing control laws and for future launcher pre-development phases. In these examples, we have shown that load performance was improved with respect to pre-existing non-structured controllers. Simplifications in the design process were highlighted, providing flight control development cost reduction perspectives. These results pave the way for the industrial application of a structured H_∞ framework for fast retuning of existing controllers and development of new controller structures for future launchers

Nomenclature

AG	(Ariane Group)
α	(Angle of attack)
$\Delta\beta_c$	(Commanded deflection around nominal deflection)
$\Delta\beta_R$	(Realized deflection around nominal deflection)
CNES	(*Centre National d'Etudes Spatiales*)
D	(Drag in body axis)
F	(Aerodynamic Center)
G	(Center of Gravity)
HF	(High Frequency)
hp_{GYi}	(ith bending mode slope at gyrometer location)
hp_{IMUi}	(ith bending mode slope at IMU location)
htu_i	(ith bending mode deformation at nozzle rotation point)
I	(Launcher pitch inertia)
I/F	(Interface)
IMU	(Inertial Measurement Unit)
ξ_i	(ith bending mode damping)
ξ_β	(Damping of actuator model)
L	(Lift in body axis)
LF	(Low Frequency)
LMI	(Linear Matrix Inequality)
LQG	(Linear Quadratic Gaussian)
LPV	(Linear Parameter Varying)
LTI	(Linear Time Invariant)
L_F	(Distance between CoG and Centre of Pressure)
l_{tu}	(Position of nozzle rotation point with respect to the launcher CoG)
m	(Launcher total mass)
Mb_i	(ith mode nozzle rotation point slope contribution)
MCI	(Mass Balance and Inertia Data)
MDB	(Model DataBase)
MIMO	(Multi-Input Multi-Output)
MISO	(Multi-Input Single-Ouptut)
P_c	(Commanded thrust)
q_i	(ith bending mode generalized coordinates)
R_c	(Barycentric reference frame)
R_l	(Body reference frame)
R_t	(Terrestrial reference frame)
$\Delta\theta$	(Launcher pitch angle deviation with respect to commanded angle)
SISO	(Single-Input Single-Output)
S_{ref}	(Reference area)
T	(Thrust in body axis)
V_R	(Relative velocity)
ΔW	(Wind perturbation)
ω_i	(ith bending mode pulsation)
ω_β	(Pulsation of actuator model)
$\Delta\dot{z}$	(Launcher lateral velocity)

References

[1] G. STEIN - *Respect the Unstable*. IEEE Control Systems Magazine, 23(4): 12-25, Aug. 2003.

[2] J. V. BURKE, D. HENRION, A. S. LEWIS, M. L. OVERTON - *HIFOO – A Matlab Package for Fixed Order Controller Design and H-Infinity Optimization*. Proc. IFAC Symp. Robust Control Design, Toulouse, France, 2006.

[3] P. APKARIAN, D. NOLL - *Nonsmooth H_∞ Synthesis*. IEEE Transactions on Automatic Control, 51(1): 71-86, 2006.

[4] P. APKARIAN, D. NOLL - *Nonsmooth Optimization for Multiband Frequency-Domain Control Design*. Automatica Journal, vol. 43, no. 4, p. 724-731, 2007.

[5] J. S. ORR *et. al.* - *Space Launch System Ascent Flight Control Design*. AAS GNC Conference, 14-038, 2014.

[6] B. WIE, W. DU, M. WHORTON - *Analysis and Design of Launch Vehicle Flight.* AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2008.

[7] J. C. DOYLE, K. GLOVER, P. P. KHARGONEKAR, B. A. FRANCIS - *State-Space Solutions to Standard H2 and H_∞ Control Problems*. IEEE Transactions on Automatic Control, 34(8), August 1989.

[8] P. A. IGLESIAS, K. GLOVER - *State-Space Discrete Time H_∞ Control Theory.* ECC 91, Grenoble, France, July 1991.

[9] P. GAHINET, P. APKARIAN - *A Linear Matrix Inequality Approach to H_∞ Control*. International Journal of Robust and Nonlinear Control, Vol 4: 421-448, 1994.

[10] P. GAHINET, P. APKARIAN - *Structured H_∞ Synthesis in MATLAB*. Proc. 18th IFAC World Congress, Milano, August 2011.

[11] M. GANET-SCHOELLER - *Towards Structured H_∞ Synthesis for Flexible Launcher*. CEAS, 2015.

[12] P. GAHINET, P. APKARIAN - *Automated Tuning of Gain-Scheduled Control Systems.* Proc. IEEE Conf. on Decision and Control, 2740-2745, December 2013.

[13] H. LHACHEMI, D. SAUSSIE, G. ZHU - *A Structured H_∞-Based Optimization Approach for Integrated Plant and Self-Scheduled Flight Control System Design*. Aerospace Science and Technology 45, 30-38, 2015.

[14] S. SKOGESTAD, I. POSTLETHWAITE - *Multivariable Feedback Control: Analysis and Design.* 2nd ed., Wiley, 2005.

[15] *Matlab Robust Control Toolbox (RCT)*. 2013.

AUTHORS

Martine Ganet-Schoeller is currently a flight control engineer at Ariane Group in Les Mureaux, France.

Clément Combier: ISAE student.

Jean Desmariaux is currently a flight control engineer at CNES Launchers Directorate in Paris, France.