Progress in Flight Dynamics, GNC, and Avionics **6** (2013) 209-228
DOI: 10.1051/eucass/201306209

VEGA ROLL AND ATTITUDE CONTROL SYSTEM ALGORITHMS TRADE-OFF STUDY

N. Paulino[1], G. Cuciniello[2], I. Cruciani[3], F. Corraro[2], D. Spallotta[3], and F. Nebula[2]

[1]GMV, P.T.M
Tres Cantos, Madrid, Spain
[2]CIRA
Via Maiorise, Capua 81043, Italy
[3]ELV
22 Corso Garibaldi, Colleferro 00034, Italy

This paper describes the trade-off study for the selection of the most suitable algorithms for the Roll and Attitude Control System (RACS) within the FPS-A program, aimed at developing the new Flight Program Software of VEGA Launcher. Two algorithms were analyzed: Switching Lines (SL) and Quaternion Feedback Regulation. Using a development simulation tool that models two critical flight phases (Long Coasting Phase (LCP) and Payload Release (PLR) Phase), both algorithms were assessed with Monte Carlo batch simulations for both of the phases. The statistical outcomes of the results demonstrate a 100 percent success rate for Quaternion Feedback Regulation, and support the choice of this method.

1 INTRODUCTION

The attitude control system for pointing and tracking capabilities of a generic spacecraft boasts many solutions deeply investigated in literature. In the case of VEGA launcher, large angle manoeuvres can be requested by the guidance function, for example, during PLR phases of the liquid propelled upper stage. The subsystem in charge of facing such features is the RACS, which is a blowdown hydrazine propelled system dealing with roll control during VEGA SRMs (Solid Rocket Motors) and LPS (Liquid Propellant System) propelled phases as well as 3-axis attitude control during transferring manoeuvres and PL release phases. Thrusters numbering and positions are reported in Fig. 1. As shown, the RACS is composed of 4 Roll Control Thrusters (RCT) — *1*, *2*, *4*, and *5* — in charge of both roll and pitch control, and 2 Longitudinal Control Thrusters (LCT) — *3*

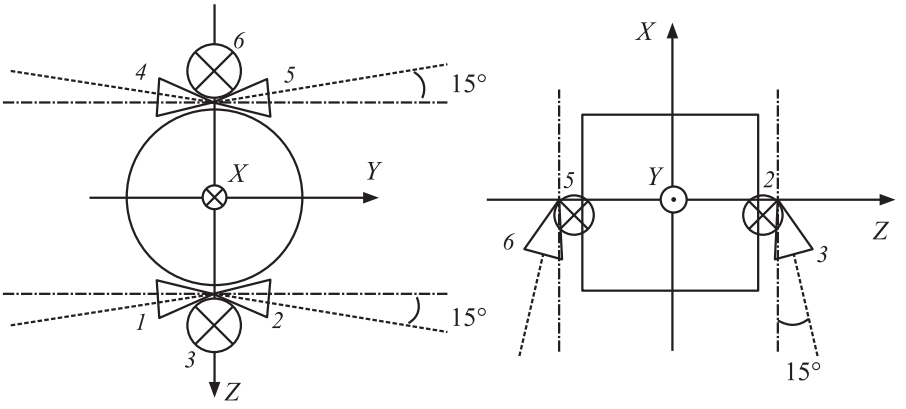

Figure 1 VEGA launch vehicle RACS geometry and thrusters numbering

and *6* — in charge of yaw control. All thrusters are characterized by a 15 degree mounting angle between thrusters' nozzles axes and body axes.

On the purpose of developing the new Flight Program Software for the VEGA launcher within the FPS-A program, a trade-off study has been considered necessary to select the most suitable RACS algorithm. Two logics have been considered as a possible candidate: SL and Quaternion Feedback Regulator (QFR). The former was originally a modification of the SL algorithms for the LYRA program ACS control system. It has been considered as a starting point for the development of the new FPS RACS algorithms, with particular concern on tuning procedure improvements. The latter is a kind of natural solution for facing single and rapid manoeuvres about the eigen axis, once gyroscopic cross coupling effect, generally introduced by the vehicle dynamics, is compensated in order to make the actuation system decoupled for each axis.

The performed trade-off is aimed at highlighting both algorithms performance with regard to a set of upper level requirements. These are listed in Table 1. They provide limitations on attitude and attitude rates during specific manoeuvres (PLR and coasting phases) depending on the PL control methodology (longitudinally spun or 3-axis control), as well as limitations due to the hardware (number of open/close cycles) during the mission. In particular, $|\varphi|$, $|\psi|$, and $|\Delta\theta|$ represent modulus of roll, yaw, and pitch angle errors respectively, while $|\Delta P|$ and $\Delta\omega_T$ denote roll rate error modulus and transversal attitude rate error. Also, hydrazine consumption and number of activations per thrusters are important parameters, since they assess both RACS control performance and mission accomplishment. Table 1 also provides relationship between considered flight phases and operational benchmark as defined in section 4.

In this section, the requirements to be met by RACS were presented. Section 2 describes the simulation tool and models. Section 3 provides a short

Table 1 Applicable requirements vs. PL and flight phase

PL type	Benchmark	Applicable requirements			
		Accuracy	Activations per thrusters	Mass consumption	Duty cycles
3-axis controlled PL during PL release phases	STAB	$\lvert\varphi\rvert < [0.75]^\circ$ $\lvert\Delta\psi\rvert, \lvert\Delta\theta\rvert < [0.5^\circ\ 0.5^\circ]$ $\lvert\Delta\omega\rvert < [0.8\ 0.3\ 0.3]$ deg/s	Total number of activations < 2000	Consumed hydrazine < 32 kg	$t_{on} > 20$ ms $t_{off} > 40$ ms
Spun PL during PL release phases	SPIN	$\lvert\Delta P\rvert < 1$ deg/s $\Delta\omega_T < 1.5$ deg/s			
3-axis controlled PL during transferring manoeuvres	LCP	$\lvert\Delta\psi\rvert, \lvert\Delta\theta\rvert < 20^\circ$ $\lvert\Delta P\rvert < 1$ deg/s			

explanation of the two algorithms and tuning procedure. Section 4 provides the trade-off description. Section 5 reports some Monte Carlo results. Section 6 synthesizes results and reports main conclusions on the performed trade-off.

2 ROLL AND ATTITUDE CONTROL SYSTEM DESIGN TOOL SIMULATOR

The RACS Design Tool is the simulator adopted for the RACS algorithm design process, for a preliminary performance and robustness assessment, and for the requirement verification during the abovementioned trade-off.

In particular, it is a three-degree-of-freedom (3DoF) simulator, implemented in a MATLAB®/Simulink® environment, able to simulate the launch vehicle (LV) rotational dynamics during specific flight phases within the VEGA launcher mission, both coasting and propelled phases. The simulator functional architecture is depicted in Fig. 2, followed by a brief explanation of all functional blocks.

2.1 Rotational Dynamics

It implements the well-known Euler's equations [1–4] describing the rotational motion of a rigid body about body-fixed axes with origin at the center of mass:

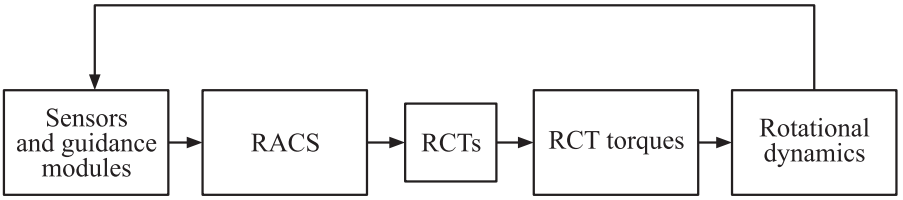

Figure 2 Simulation tool architecture of RACS

$$I\dot{\omega} = \Omega I\omega + u$$

where ω is the angular velocity vector; u is the control torque vector; I is the inertia matrix; and Ω the skew-symmetric matrix defined by

$$\Omega = -\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} . \tag{1}$$

Note that the only parameter is the inertia matrix. The block also implements the quaternion kinematics:

$$\dot{q} = \frac{1}{2}\,\Omega q + \frac{1}{2}\,q_4\omega\,; \quad \dot{q}_4 = -\frac{1}{2}\,\omega^{\mathrm{T}} q$$

where $q = [q_1, q_2, q_3, q_4]^{\mathrm{T}}$ is the quaternion vector describing the body rigid dynamics, with first three components being the vector part and fourth scalar component being related to the magnitude of the Euler axis rotation.

2.2 Thrusters Torques

Due to a 15 degree mounting angle between thrusters nozzles axes and body axes (see Fig. 1), each thruster provides two thrusts components: the former aligned with its mounting body axis (y axis for *1*, *2*, *4*, and *5* thrusters and x axis for thrusters *3* and *6*), the latter aligned with the z axis. Thus, when computing control torques due to thrusters action, the torques about the x, y, and z axes can be expressed by:

$$\begin{aligned}
M_x^{\mathrm{thrust}} &= R\,(-T_1^{\cos} + T_2^{\cos} + T_4^{\cos} - T_5^{\cos})\,; \\
M_y^{\mathrm{thrust}} &= b\,\left(T_1^{\sin} + T_2^{\sin} + T_3^{\sin} - T_4^{\sin} - T_5^{\sin} - T_6^{\sin}\right) + R\,(T_3^{\cos} - T_6^{\cos})\,; \\
M_z^{\mathrm{thrust}} &= b\,(T_1^{\cos} - T_2^{\cos} + T_4^{\cos} - T_5^{\cos})
\end{aligned}$$

where R is the LV radius at the height of the RACS clusters; b is the longitudinal distance between LV center of gravity (CoG) and RACS clusters: it is positive

if CoG is below the thrusts application points (with respect to the positive sense of the longitudinal axis), negative otherwise; $T_i^{\cos} = T_i \cos(15°)$ and $T_i^{\sin} = T_i \sin(15°)$ for $i = 1, \ldots, 6$.

2.3 Roll and Attitude Control System Performance Model

The RACS performance model implements a set of equations describing the functioning of a hydrazine propelled blowdown system, with pressure (and so thrust and mass flow) proportionally varying with residual hydrazine.

In particular, the implemented equations represent each thruster's mass flow, thrust and temperature dynamics, as well as tank residual hydrazine resulting from activations commands profiles received from the guidance module. It is based on thrusters Hardware (HW) model by the Rafael company, feeded by a nominal hydrazine mass of 37.5 kg at 24.7 bar. The performance model is composed of a first equation providing the link between hydrazine tank residual mass and tank pressure (Fig. 3*a*):

$$p_{\text{tank}} = p_{\text{tank0}} \left(\frac{m_{\text{N}_2\text{H}_4,0} - a_5 K}{m_{\text{N}_2\text{H}_4} - a_5 K} \right)^{1/a_{10}}$$

with $K = a_4 [1 + a_{27} (p_{\text{tank0}} - a_{28})]$, $m_{\text{N}_2\text{H}_4,0}$ and p_{tank0} representing initial hydrazine mass and initial tank pressure, and a_4, a_5, a_{10}, a_{27}, and a_{28} proper empirical coefficients chosen to fit experimental data; a set of equations describing the injection pressure at each thruster (p_{feed}) as function of the tank pressure p_{tank} and of the number of simultaneously activated thrusters, whose coefficients

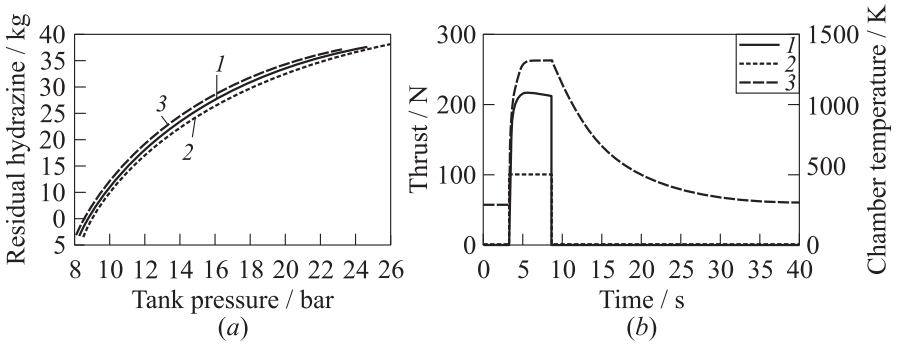

Figure 3 Residual hydrazine vs. tank pressure (*a*) (*1* — nominal; *2* — maximal; and *3* — minimal) and thrust and temperature within a 5-second pulse (*b*) (*1* — thrust; *2* — scaled command; and *3* — temperature)

are computed basing on the piping characteristics; and, finally, a set of differential equations describing trends of thrust, mass flow, and temperature dynamic for each thrusters (Fig. 3*b*), also accounting for opening and closure time delay due to HW valves.

In particular, two operative modes are defined:

(*i*) the ON mode (opened valve), whose governing equations are:

$$\begin{aligned} \dot{T}_C &= c_1 p_{\text{feed}}^{c_2} \left(T_{C,\max} - T_C\right) ; \\ c_3 T_C \dot{F} &= c_6 p_{\text{feed}}^{c_7} T_C^2 - c_8 F T_C^{3/2} ; \\ \dot{m} &= c_{11} p_{\text{feed}}^{c_{12}} \end{aligned}$$

where $T_{C,\max} = c_9 p_{\text{feed}}^{c_{10}}$; and

(*ii*) the OFF mode (closed valve), defined by

$$\begin{aligned} \dot{T}_C &= -c_{13} \left(T_C - T_{\text{env}}\right) ; \\ \dot{F} &= -F \frac{c_{18}\Omega}{[1 + \Omega(t - t_{\text{off}} - \Delta t_{\text{delay off}})]} ; \\ \dot{m} &= 0 \end{aligned}$$

where

$$\Omega = \frac{c_{19} c_{14} \sqrt{T_C}\, p_{\text{feed}}^{c_{15}}}{1 + c_{16} (\Delta t_{\text{on}} + \Delta t_{\text{delay off}} - \Delta t_{\text{delay on}})^{-c_{17}}} = c_{19} c_{14} \sqrt{T_C}\, p_{\text{feed}}^{c_{15}}$$

being $c_{16} = 0$; T_C is the generic thruster's chamber temperature (in K); F is the related thrust (in N); $\dot{m}$ is the mass flow (in kg/s); p_{feed} is the injection pressure (in Pa); T_{env} is the environment temperature (in K); and $\Delta t_{\text{delay on}}$ and $\Delta t_{\text{delay off}}$ are the delays on opening and closure times. Also, in this case, all constants c are the empirical scalar parameter properly chosen to fit experimental data.

Finally, the model also provides the possibility to scatter a set of parameters: initial hydrazine mass, initial tank pressure, thrust, and opening valve time delay. It also accounts for thrust efficiency degradation due to atmospheric pressure.

2.4 Sensor and Guidance Module

This module simulates the guidance and navigation functions. Therefore, it provides to the RACS Algorithm Module both the commanded and the measured quaternion, adding a measurement noise consistent with the LV vibratory environment.

2.5 Roll and Attitude Control System Algorithms Module

This module is the core of the RACS Design Tool and includes both RACS algorithms described in the following sections, implemented in SIMULINK® or m-file S-functions blocks.

3 ALGORITHMS DESCRIPTION

3.1 Switching Lines

This algorithm computes the actuation time axis by axis, assuming a decoupling between the 3 axes. For each axis, respective thrusters opening time is computed. In the case of roll and pitch control, a concurrence on the thrusters pairs takes place. This conflict is resolved and saturation is applied to each outputted opening time to assure minimum opening and closing times. The parametrization in the phase plane of the dead-zone and dead-band is calculated from the mission data and the tuning to obtain the parameters for the switching lines. If the state is outside of the dead-band and dead-zone ($S \neq 0$), then the error is computed, and so is the opening times needed for the correction. The actuation time is evaluated based on an estimation of the torque/angular velocity variation relationship. In particular, opening time is computed using the nominal inertia and arm values, as well as the estimated available thruster's force. Finally, the opening time is quantified before processed by the architecture medium level.

A graphical and conceptual explanation of SL method is depicted in Fig. 4. The letters define working regions. White region defines a dead-zone and dead-band, and the bold dashed line in Fig 4*b* indicates the reference $x_{2\text{desired}}$ for the angle velocity.

The algorithms are based on SL logic over the phase plane for a single axis. For each channel (roll, yaw, and pitch) the state space (phase plane) is divided in four regions — A, B, C, and D. At each control step, the RACS receives the attitude errors and the angular rate errors and computes the ignition zones. If the LV state is in the zone where $S = 0$, then the thrusters will be switched off (since it is the dead-zone or the dead-band). The decision lines are parameterized by the quintuplet $\{A, B, T_1, T_2, x_0\}$:

$$\begin{cases} \alpha_1 : x_2 T_1 = -x_1 + x_0 \,; \\ \alpha_2 : \quad x_2 = B \,; \\ \alpha_3 : \quad x_1 = A \,; \\ \alpha_4 : x_2 T_2 = -x_1 + A \,; \end{cases} \qquad \begin{cases} \beta_1 : x_2 T_1 = -x_1 - x_0 \,; \\ \beta_2 : \quad x_2 = -B \,; \\ \beta_3 : \quad x_1 = -A \,; \\ \beta_4 : x_2 T_2 = -x_1 - A \,. \end{cases}$$

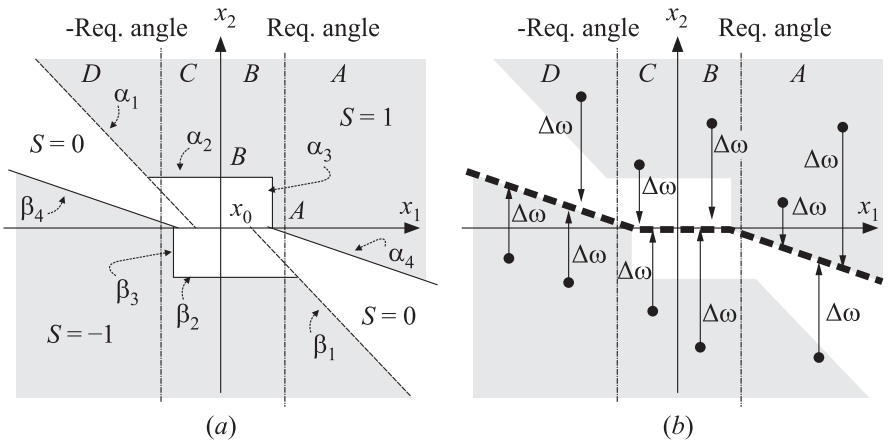

Figure 4 Switching lines in the phase plane (a); and computation of the angular velocity error (b)

The actuation signal regions are set by the logic:

$$\begin{aligned} ((\alpha_1 \leq 0) \wedge (\alpha_2 < 0)) \vee ((\alpha_3 \leq 0) \wedge (\alpha_4 < 0)) &\Rightarrow S = 1\,; \\ ((\beta_1 \geq 0) \wedge (\beta_2 > 0)) \vee ((\beta_3 \geq 0) \wedge (\beta_4 > 0)) &\Rightarrow S = -1\,; \\ \text{otherwise} &\Rightarrow S = 0\,. \end{aligned}$$

The angular error is computed comparing the distance between the velocity error and specific lines in the phase plane (bold dashed lines in Fig. 4b). This definition of the error aims at forcing the state into a dead-band and dead-zone as long as possible to save actuation and firing. The actuation is given by the opening time Δt computed using the inversion of the dynamics:

$$J_0^{-1} \begin{bmatrix} 2\hat{F}_{SS} L_0 \Delta t_x \\ \hat{F}_{SS} L_0 \Delta t_y \\ 2\hat{F}_{SS} L_{\text{pitch0}} \Delta t_z \end{bmatrix} = \begin{bmatrix} \Delta\omega_x \\ \Delta\omega_y \\ \Delta\omega_z \end{bmatrix}$$

$$\left(\varepsilon_x = \frac{2\hat{F}_{SS} L_0}{J_{xx0}}\,; \quad \varepsilon_y = \frac{\hat{F}_{SS} L_0}{J_{yy0}}\,; \quad \varepsilon_z = \frac{2\hat{F}_{SS} L_{\text{pitch0}}}{J_{zz0}} \right)$$

where $\hat{F}_{SS}$ is the estimated force outputted by the RCT (estimated from the mass using a polynomial filter); J_0 is the nominal inertia matrix (assumed dominated by its diagonal elements); L_0 is the nominal arm values to compute the torque in x and y axis; and L_{pitch0} is the nominal arm value to compute the torque in the z-axis.

For region A (and similar to region D), the actuation time depends indirectly on the angle (in the reference angular velocity computation), introducing

Table 2 Actuation time computation for the different regions

Region	Actuation time
A	$\Delta t = \dfrac{\Delta\omega}{\varepsilon} = \dfrac{x_{2\text{desired}} - x_2(k)}{\varepsilon}$ $\left(\text{with } x_{2\text{desired}} = \dfrac{A - x_1(k)}{T_2}\right)$
B and C	$\Delta t = K_a \dfrac{\Delta\omega}{\varepsilon} = K_a \dfrac{x_{2\text{desired}} - x_2(k)}{\varepsilon}$
D	$\Delta t = \dfrac{\Delta\omega}{\varepsilon} = \dfrac{x_{2\text{desired}} - x_2(k)}{\varepsilon}$ $\left(\text{with } x_{2\text{desired}} = \dfrac{-A - x_1(k)}{T_2}\right)$

a proportional-derivative (PD) control behavior. For regions B and C, the actuation time is computed directly inverting the dynamics and a gain K_a that is used to introduce an error scaling (Table 2). Note that A and D use the dead-band to update the reference; C and B use directly the velocity to invert the dynamics and push the velocity to zero. This algorithm aims at using as most as possible the phase plane stable regions, and uses the dynamics of the system to steer the error to an acceptable region.

Moreover, the use of dead-band and dead-region allows saving actuations for small errors, and prevents chattering. In regions A and D, the use of the phase plane allows separating the response into three separate phases: capture, convergence, and limit cycle (Fig. 5).

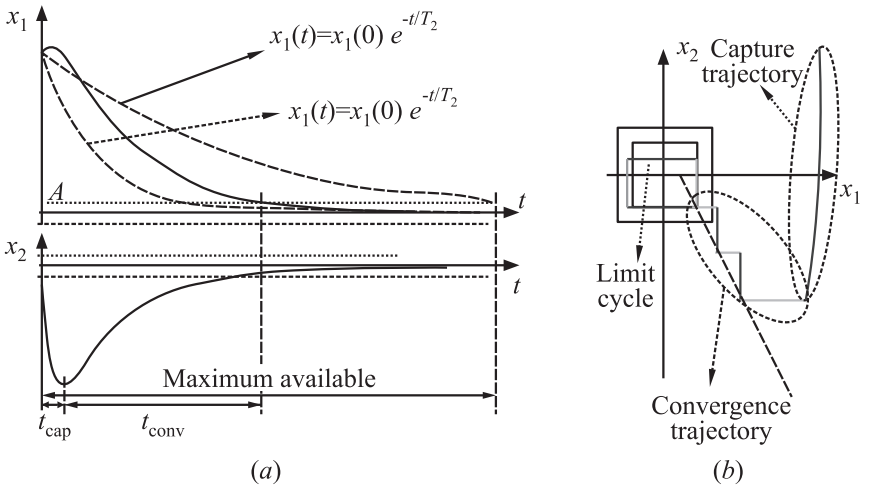

Figure 5 Desired behavior (a) (t_{cap} — capture time and t_{conv} — convergence time); and relationship with SL parameterization (b)

The capture phase reaches the dead-band/dead-zone as fast as possible, with full steady-state actuation (maximum opening time). During convergence time, the angle follows approximately an exponential behavior where the time constant is limited by T_1 and T_2. If T_2 is minimum (corresponding to the narrowest dead-band), the resulting convergence behavior approximates a PD controller with dead-band, with $T_1 = K_d/K_p$. The response will be the fastest, at the expense of more activation and possible chattering. Time T_2 should be chosen as the slowest possible settling time that reaches the requirements within the necessary time ($\phi_{\max}(0)$ is the maximum possible initial angle and A is the dead-zone limit that the state should reach):

$$T_2 = \frac{\text{Maximum set time}}{\ln(\phi_{\max}(0)) - \ln(A)}\,. \tag{2}$$

If the state is in regions C and D, but outside the dead-zone, the actuation depends only on the velocity error (since the angle requirements are already met): $\Delta t = K_a \Delta\omega/\varepsilon$.

When the dead-region is reached (requirements fulfilled), a limit cycle is induced to reduce actuation while the state is keeping the requirements. The parameter limit_cycle is used as reference velocity during this phase.

If the inertia is perfectly known, K_a can be set to 1. In the presence of uncertainty on the dynamics, the parameter K_a introduces conservativeness in the computation of the actuation ($K_a < 1$) to avoid overshooting the velocity state.

Table 3 sums up the tunable parameters that influence the control laws. These are tuned for each phase and depend on the requirements and mission data.

Table 3 Tunable parameters

Tuning	STAB pointing (B1)	SPIN (B3)	STAB LCP (B3)
T_1	$\frac{K_d}{K_p}$ ($K_d = \frac{\text{req.angle}}{\text{req.rate}}$, $K_p = 1$)		
T_2	$\frac{\text{Maximum set time}}{\ln(\phi_{\max}(0)) - \ln(A)}$	$[0,0,0]$ (no impact)	$\frac{\text{Maximum set time}}{\ln(\phi_{\max}(0)) - \ln(A)}$
limit_cycle	$\frac{\text{req.vel}}{2}$ (if control resolution is too high for requirement, $2\,\text{req.vel} > \Delta t_{\min}\varepsilon_{\max} > \text{req.vel}$)	0 ($\text{req.vel} > \Delta t_{\min}\varepsilon_{\max}$)	
K_a	$K_{a_x} < \frac{J_{x_{\min}}}{J_{x_{\text{nom}}}}$, $K_{a_y} < \frac{J_{y_{\min}}}{J_{y_{\text{nom}}}}$, $K_{a_z} < \frac{J_{z_{\min}}}{J_{z_{\text{nom}}}}$		

3.2 Quaternion Feedback Regulator

The functional architecture of the QFR algorithm is depicted in Fig. 6.

First of all, the angular velocity of the launcher (ω_f) is computed starting from an online estimation of the related quaternion rate. This estimation is carried out using a high pass second-order filtering, tuned in function of the noise characterization. On the other hand, a discrete derivative is applied on the reference quaternion (q_d), coming from the Guidance module in order to obtain the desired programmed angular velocity (ω_d).

The so-computed angular velocities, with the quaternion error q_e (between the programmed and the launcher frames), feed a downstream block dedicated to the reduction of the hydrazine mass consumption. In order to save mass, the thrusters are not activated when the error is within the required performances with some margins. More in detail, the controller is enabled only if the angular velocities' errors or the prediction of the attitude by means of a linear extrapolation are outside a requirement region with some margins.

The core of the QFR algorithm is the dynamic inversion control law used for computing the desired torque command. More in depth, the torque command vector is computed simultaneously on the three axes as

$$M_e = -G\,(\Omega I \omega_f) - D\,(\omega_f - \omega_d) - K q_e$$

where G is the Boolean that enables or disables the gyroscopic coupling compensation ($\Omega I \omega$); Ω is the skew-symmetric matrix of Eq. (1); I is the launcher inertia matrix; ω_f is the launcher angular velocity; and ω_h is the manipulation of ω by a proper threshold function.

It is observed that in Eq. (2), the second term introduces an artificial damping factor (*derivative action*) with a feedforward action (ω_d), while the last term

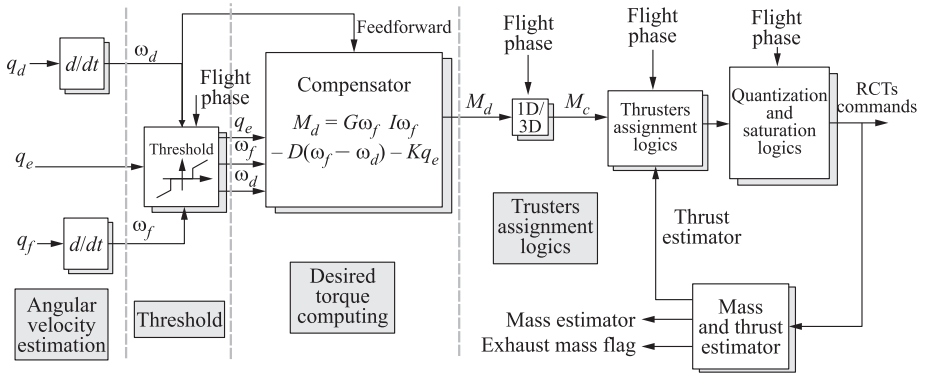

Figure 6 Quaternion feedback regulator functional architecture

is a linear quaternion error feedback (*proportional action*). The weights of these control actions are related to D and K and they are full matrices taking into account also the inertia products of I. For this reason, the QFR algorithm assumes MIMO (multiple input multiple output) controller characteristics. In conclusion, the abovementioned matrices shall be tuned — as a function of the launcher inertia characteristics — in order to confer a closed loop equivalent second-order behavior characterized by the desired natural frequency and damping [5].

Moreover, it is worth noting that the commanded torques about yaw and pitch axes are set to zero if only roll rate control is requested, in order to minimize interferences with the attitude controller during the propelled phases of flight. This reconfiguration is performed by the block "1D/3D" downstream of the above described control law.

Finally, the thrusters and the vehicle dynamics are equivalent to a system reacting only to the medium value of the imposed torque and this allows the implementation of the commanded torque as a Pulse Width Modulated (PWM) signal.

For what concern the thrusters assignment logics, from a general point of view, given the thrusters mounting configuration (see Fig. 1), the launcher CoG position, and an estimation of the maximum steady-state force provided by a single thruster (still modeled with an interpolation function of the remaining hydrazine mass), it is possible to compute the PWM duty-cycle by means of the opening time required for each thruster, starting from the desired torque.

The particular VEGA configuration with 6 thrusters allows considering only three thrusters in order to produce the desired torque about each launcher axis (indeed, the remaining three thrusters perform opposite rotations). For example, using the thrusters *1*, *3*, and *4* as reported in Fig. 1, the thruster opening times computing problem is reduced to the solution of a 3×3 system of linear equations, starting from the desired torque vector. The system solution is then processed by an algorithm in order to obtain feasible duty cycles by means of positive actuation times. In particular, in the case of a negative solution, the relative thruster is substituted with the *dual* one also compensating the residual torque due to the mounting angle about the second axis using the thrusters *3* or *6*. The complete algorithm that is composed by three steps is reported in Table 4 where N_1, N_2, and N_3 are the solutions of the system of linear equations and represent the opening time.

The computed thruster opening times are real numbers. Since they shall be integer because of the cycle time of the controller, an approximation to the nearest integer is needed to obtain integer numbers. Other constrains about the minimum opening/closing times come directly from the avionics requirements and are applied to the quantization and saturation logics as well as the activation policy when the opening time is less than the allowable minimum or is greater than the controller cycle time. Finally, although the great advantages by means of robustness with reference to the external torque disturbances and to the thrusters uncertainties, the thrust estimation algorithms reported in the architecture of

Table 4 Procedure for the opening times computation

Check	First Step		Second Step	Third Step	
	Action 1	Action 2	Action*	Action 1	Action 2
$N_1 < 0$	$N_2 = -N_1$	$N_1 = 0$	$N_6 = N_6 + 2R_{Y13}N_2$		
$N_3 < 0$	$N_6 = -N_3$	$N_3 = 0$			
$N_4 < 0$	$N_5 = -N_4$	$N_4 = 0$	$N_3 = N_3 + 2R_{Y13}N_5$		
$N_3 > N_6$				$N_3 = N_3 - N_6$	$N_6 = 0$
$N_6 > N_3$				$N_6 = N_6 - N_3$	$N_3 = 0$

*R_{Y13} is the torque contribution ratio between thrusters *1* and *3* on the Y axis.

Fig. 6 are not implemented in the current RACS software revision because of some constraints at system level. In particular, the torque estimation algorithm suffers of delays and noise due to a double quaternion differentiation and a very low sampling frequency for the computation (5 Hz) with respect to the thrusters bandwidth (about 10 rad/s); so, the tuning and the implementation of this kind of algorithm result are very difficult and do not give the expected benefits.

4 TRADE-OFF ANALYSIS

Standing to the evolution of the SL logic with respect to the initially presented version and the proposal of QFR as an alternative control solution, a comparison analysis between the two algorithms has been considered necessary on the purpose of selecting the baseline for the next developing phase. Such a trade-off analysis has been performed by comparing algorithms in term of the following parameters: control accuracy, number of activations and consumption, missionization complexity and robustness, implementation complexity, and appropriate justification of design.

Operational and Simulation Environments

To evaluate algorithms performance, attention has been paid on two flight phases considered to be dimensioning for different RACS functions: the PLR which is the phase in charge of PL injection into the foreseen orbit and the LCP, with a low rate spin, representing the flight during orbit transferring.

As a result, three proper benchmark scenarios have been defined in accordance with the considered flight phases: the SPIN, the STAB, and the LCP.

During the SPIN, a spinning PLR is considered. The PL is spun to a desired roll rate while keeping the transversal angular rate limited. The gyroscopic effect could become relevant in such a case.

Table 5 Nominal MCI of the considered PLs

Payload		Benchmark	
		STAB	SPIN, LCP
A	Mass, kg	300	699.68
	x_{CoG}, m	1.1 + 23.811	24.65
	J_{Pr}, kg·m^2	[230 325 325]	[86.94 94.51 104]
B	Mass, kg	300	300
	x_{CoG}, m	0.1 + 23.811	0.1 + 23.811
	J_{Pr}, kg·m^2	[390 250 1050]	[250 250 250]
C	Mass, kg	2500	2500
	x_{CoG}, m	1.74 + 23.811	1.75 + 23.811
	J_{Pr}, kg·m^2	[1650 3375 3375]	[3375 3375 3375]

During the STAB, a 3-axis controlled PLR is considered. Payload must be pointed to a specific attitude. This case has the most strict attitude and velocity pointing requirements.

Finally, during the LCP, a 3-axis stabilization for a long period of time is considered. Despite requirements on attitude and velocity are not as strict as for the former benchmarks, minimization of the number of activations and mass consumption are very important drivers due to the phase's long duration.

Once the operational benchmarks have been set, algorithms evaluation performance needs PL characteristics definition. On such a purpose, three different PLs have been considered, with PLs MCI characteristics covering the whole PL domain. Moreover, PLs MCI varies depending on the considered benchmark. The three PLs have been labelled "A," "B," and "C," with A originally adopted for the RACS tuning development and algorithms testing, while B and C used to verify the algorithms missionization process. The three PLs MCI characteristics are summed up in Table 5.

It is worth to point out that each algorithm accounts for the adopted PL nominal MCI, i. e., its parameters are customized on the current nominal PL (A, B, or C).

As a result of PLs and benchmarks definition, trade-off consists of 9 simulations campaigns (three PLs for each of the three benchmarks), each campaign being performed for both algorithms (SL and QFR).

Within each simulation campaign, the environment was composed of the RACS Design Tool simulator (see section 2), the data parameterization for the RACS performance model (see subsection 2.3), and the algorithms implemented in MATLAB®/Simulink® pseudocode have been adopted.

As far as scattering configurations have been concerned, the scattering on MCI, thrusters performance, and initial conditions together with a constant disturbing torque in the range [−1.2, +1.2] N·m on 3 axes have been set.

Table 6 Performance comparison for STAB and SPIN cases

PL	Algorithm	Success rate	RCT mass consumption, kg			RCT number of activations		
			Mean	Std	Max	Mean	Std	Max
STAB								
A	SL	1	0.1780	0.0452	0.2996	144.74	31.5363	270
	QFR	1	0.1904	0.0472	0.3215	129.11	24.9354	189
B	SL	1	0.1774	0.0423	0.3409	146.46	30.4453	277
	QFR	1	0.1884	0.0398	0.2952	128.57	21.8984	193
C	SL	1	0.4646	0.1665	1.5120	102.61	24.4292	224
	QFR	1	0.7516	0.2112	1.2780	172.06	21.1726	232
SPIN								
A	SL	1	0.1611	0.02	0.2104	28.92	5.2217	43
	QFR	1	0.1431	0.0114	0.1751	36.69	8.0938	60
B	SL	1	0.2213	0.0194	0.2651	22.87	5.6024	36
	QFR	1	0.1752	0.0110	0.2032	31.42	5.5253	50
C	SL	1	0.9828	0.0665	1.1387	3.14	1.8092	12
	QFR	1	1.2696	0.1503	1.5449	32.35	8.4068	49

Table 7 Performance comparison for LCP case

PL	Algorithm	Success rate	RCT mass consumption, kg			RCT number of activations			Max single
			Mean	Std	Max	Mean	Std	Max total	
A	SL	0.92	2.8012	1.2526	8.8746	2469.4	1013.6	7354	2148 (2 cases)
	QFR	1	1.9065	0.5251	3.0787	1712.61	521.0668	3187	2133 (2 cases)
B	SL	0.98	7.9543	2.6257	14.3979	7485.3	1794.2	12042	3195 (115 cases)
	QFR	1	1.8514	0.5372	3.1032	1666.1	561.9088	3259	1819
C	SL	0.95	2.73	1.6338	12.2906	647.99	506.5122	4502	1471
	QFR	1	3.3570	0.8720	5.6616	2462.7	734.2555	4292	3333 (22 cases)

For each benchmark-PL case, 100 Monte Carlo runs have been performed. Table 1 points out the relationship between simulated benchmark and verified requirements.

5 TRADE-OFF RESULTS

This section reports the results of the trade-off analysis between SL and QFR algorithms. Provided data compare the algorithms in terms of performance compliance with a success rate of 1 (100 percent fulfillment), hydrazine mass consumption, and thrusters total number of activations.

The comparisons are grouped within the selected benchmarks (STAB, SPIN, and LCP) in the following subsections. In particular, numerical results concerning hydrazine consumption and total number of activations are collected in

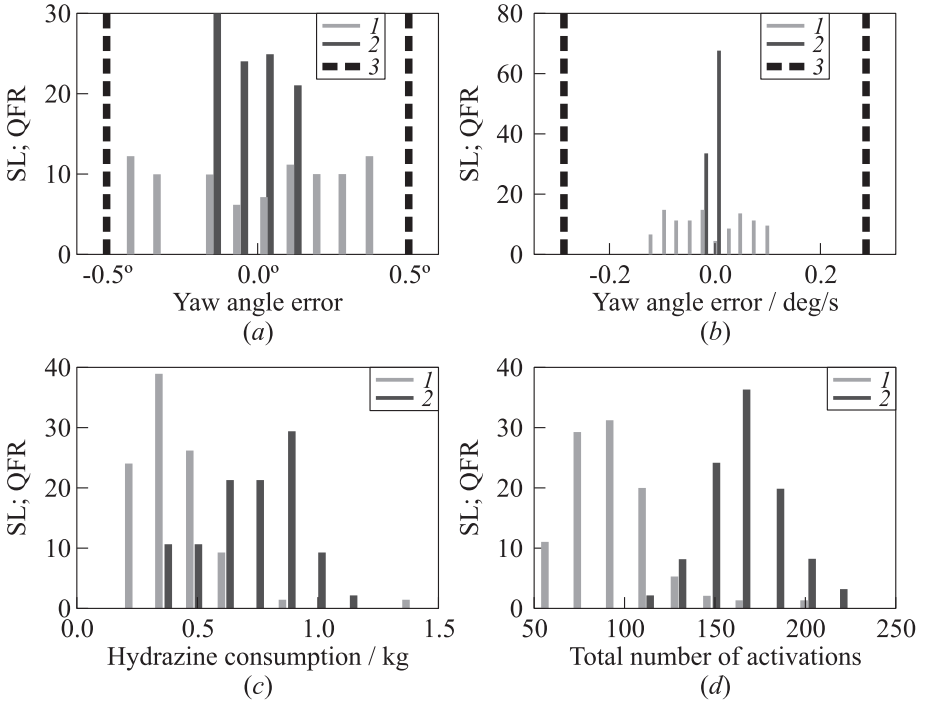

Figure 7 Relevant histograms for STAB benchmark (PL C): (*a*) yaw angle error; (*b*) yaw rate error; (*c*) hydrazine consumption; and (*d*) total number of activations: *1* — SL; *2* — QFR; and *3* — requirements

Table 6 for STAB and SPIN and Table 7 for LCP benchmarks, while Figs. 7–9 provide relevant histograms, also concerning attitude and attitude rates errors.

5.1 STAB

All requirements are met for both algorithms, for all PLs. Both algorithms present similar performance. However, following considerations can be pointed out. The QFR attitude accuracy is slightly better than SL, especially for yaw angle error, with SL providing more frequently nonzero errors, and QFR having a narrow-Gaussian behavior, as shown in Fig. 7*a*. The attitude rate accuracy evidences a quite Gaussian behavior for both algorithms. However, QFR errors are statistically smaller than for SL, especially for Q error of PL C (maximum mass) (Fig. 7*b*).

Furthermore, the QFR provides higher hydrazine consumption (higher accuracy) than SL (see Tables 6 and 7), especially for PL C, with SL mean value to be 0.46 kg with regard to 0.75 kg for QFR (Fig. 7*c*).

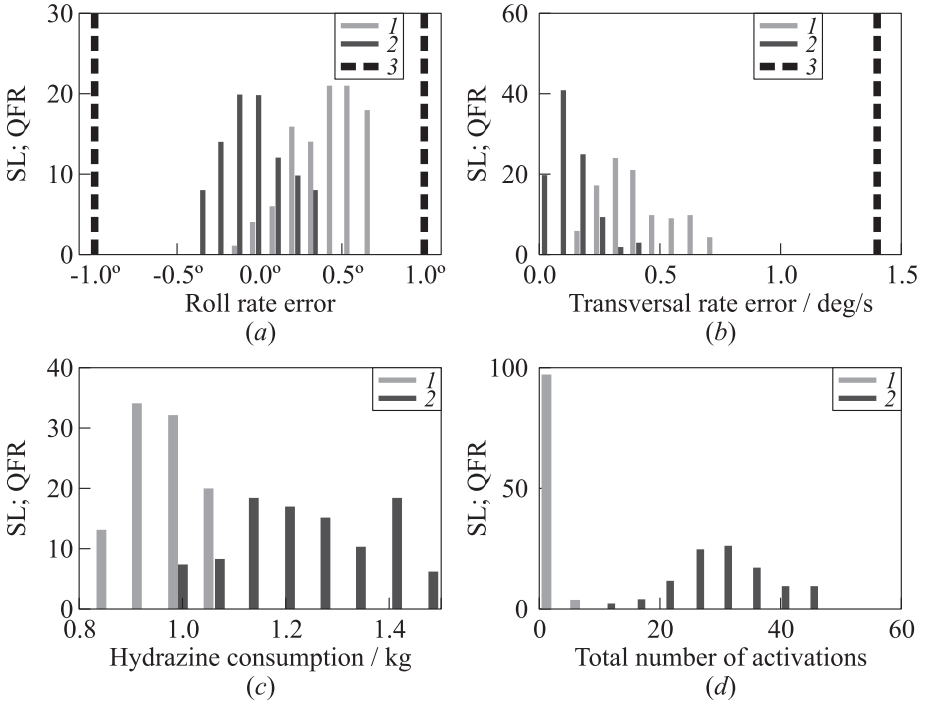

Figure 8 Relevant histograms for SPIN benchmark: (*a*) roll rate error (PL C); (*b*) transversal rate error (PL A); (*c*) hydrazine consumption (PL C); and (*d*) total number of activations (PL C): *1* — SL; *2* — QFR; and *3* — requirements

Finally, SL provides a slightly higher number of activations for PLs A and B, while for PL C, QFR results to be more dimensioning (about 180 activations) than SL (about 100 activations) (Fig. 7*d*).

5.2 SPIN

Both algorithms fulfill the requirements exhibiting almost the same control accuracy. In particular, the roll rate error accuracy is slightly better for SL except for PL C for which QRF provides better results (Fig. 8*a*). Furthermore, the transversal rate error accuracy is similar, with a relevantly better behavior of QFR for PL A (Fig. 8*b*).

Finally, even though SL provides slightly higher consumption, its performance is particularly good for PL C in terms of total number of activations and consumption (Figs. 8*c* and 8*d*).

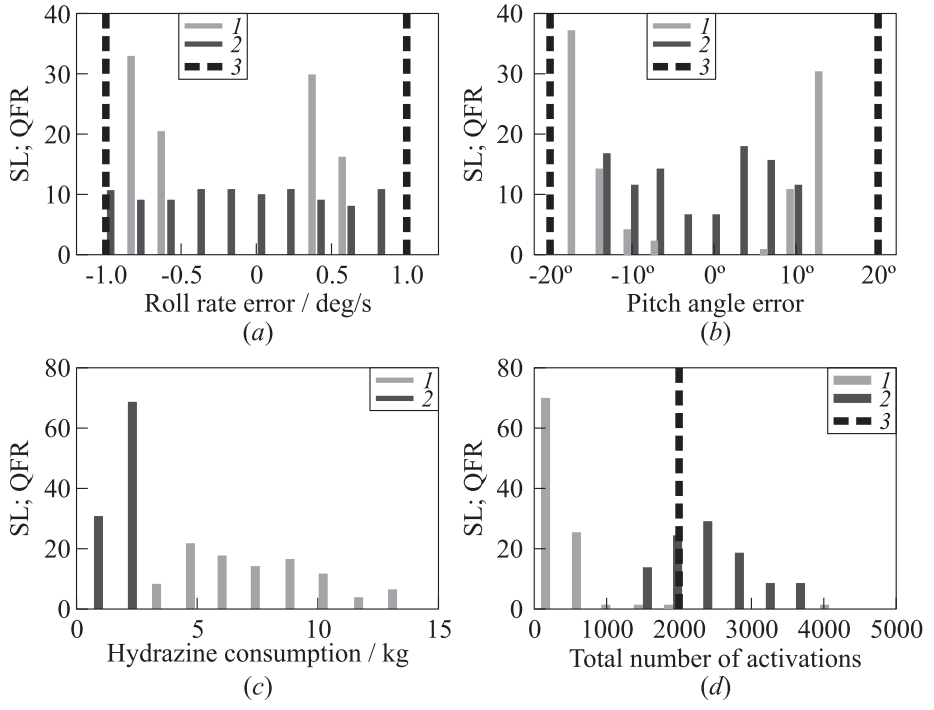

Figure 9 Relevant histograms for LCP benchmark: (*a*) roll rate error (PL C); (*b*) pitch angle errer (PL A); (*c*) hydrazine consumption (PL B); and (*d*) total number of activations (PL C): *1* — SL; *2* — QFR; and *3* — requirements

Table 8 Trade-off key results

Parameter	SL	QFR
Stability and robust stability	The algorithm switching nature makes stability more complex to be analytically demonstrated	Stability can be demonstrated by extending results from the available literature [5, 6], adopted to define controller structure, torque uncertainty, saturation, and deadzone
Performance (control accuracy)	Accuracy performances are satisfied, except for the LCP case, which experiences a significant number of noncompliances	Accuracy performances are satisfied, with higher margins with regard to limit performance than the SL logic
Performance robustness	Performance degrades significantly, specially in the presence of scattering on RCTs characteristics	This exhibits less dispersion with regard to the mean performance
Consumption and activations	For PLs B and C, LCP consumption is very high and activations are out of the requirements	Consumption does not depend significantly on PLs. Total and single RCT numbers of activations in LCP shall be further reduced to face requirements
Missionization procedure complexity	Four parameters need to be tuned, but only two of them shall be missionized with regard to PL characteristics	Five parameters shall be missionized with a given procedure that does not need any tuning optimization
Robustness of missionization	Requirements are verified, performance is similar for all PLs in STAB and SPIN cases. For LCP, performance degrades severely despite missionization	Requirements are verified. Performances are quite similar for all three PLs and cases
Implementation complexity	Controller is the same for all phases. Computations are very simple (only errors). Remaining parts are basically tables. Some interpolation is also needed during propelled phases	Implementation is quite straightforward. Computations consist of two matrix to vector products and an inversion of a 3×3 matrix. Interpolation of gains is needed during propelled phases. C-code implementation is more complex than for SL

5.3 Long Coasting Phase

The RCTs number of activations noncompliances is reported in Tables 6 and 7 and Fig. 9*d* in parentheses. For the PLs A and B, the roll rate slightly exceeds the required one (1 deg/s) during the initial time instants (see Fig. 9*a*) while for the PLs A and B, the yaw and pitch angles accuracy is not compliant for a significantly number of cases within the benchmark. Such noncompliances are, however, recovered by control action so that the final-time scenario is acceptable (see, for example, Fig. 9*b*). Finally, SL exhibits higher consumption or larger consumption dispersion (see Fig. 9*c*).

6 CONCLUDING REMARKS

Standing to provided results, SL and QFR algorithms behaviors are quite similar. However, some conclusive observations can be pointed out. First of all, the QFR algorithm generally satisfies the required accuracy performances with larger margins (higher PL injection accuracy) with regard to the SL. Furthermore, QFR exhibits a better accuracy for PLs A and B (for all benchmarks). As general consideration, in LCP, both controllers are for some cases not compliant to total number of activations requirement (mostly, for heavy PLs), but only SL is not compliant with attitude and attitude rate requirements. Finally, SL algorithm is more affected by very large RCT consumption for LCP (PL B only) and by a very large consumption dispersion for heavy PLs. Table 8 summarizes the remarks concerning the trade-off analysis.

REFERENCES

1. Lazennec, H. 1966. *Pilotage de Missiles et des Véhicules Spatiaux*. Paris: Dunod.
2. Wertz, J. R. 1980. *Spacecraft attitude determination and control*.
3. Zarchan, P. 1994. Tactical and strategic missile guidance. AIAA technical missile ser. AIAA. 297–305.
4. Sidi, M. 1997. *Spacecraft dynamics and control*. Cambridge University Press.
5. Wie, B., H. Weiss, and A. Arapostathis. 1989. Quaternion feedback regulator for spacecraft eigenaxis rotations. *J. Guidance* 12(3).
6. Guerrero-Castellanos, J. F., N. Marchand, S. Lesecq, and J. Delamare. 2008. Bounded attitude stabilization: Real-time application on four-rotor mini-helicopter. *17th IFAC World Congress Proceedings*. Seoul, Korea.