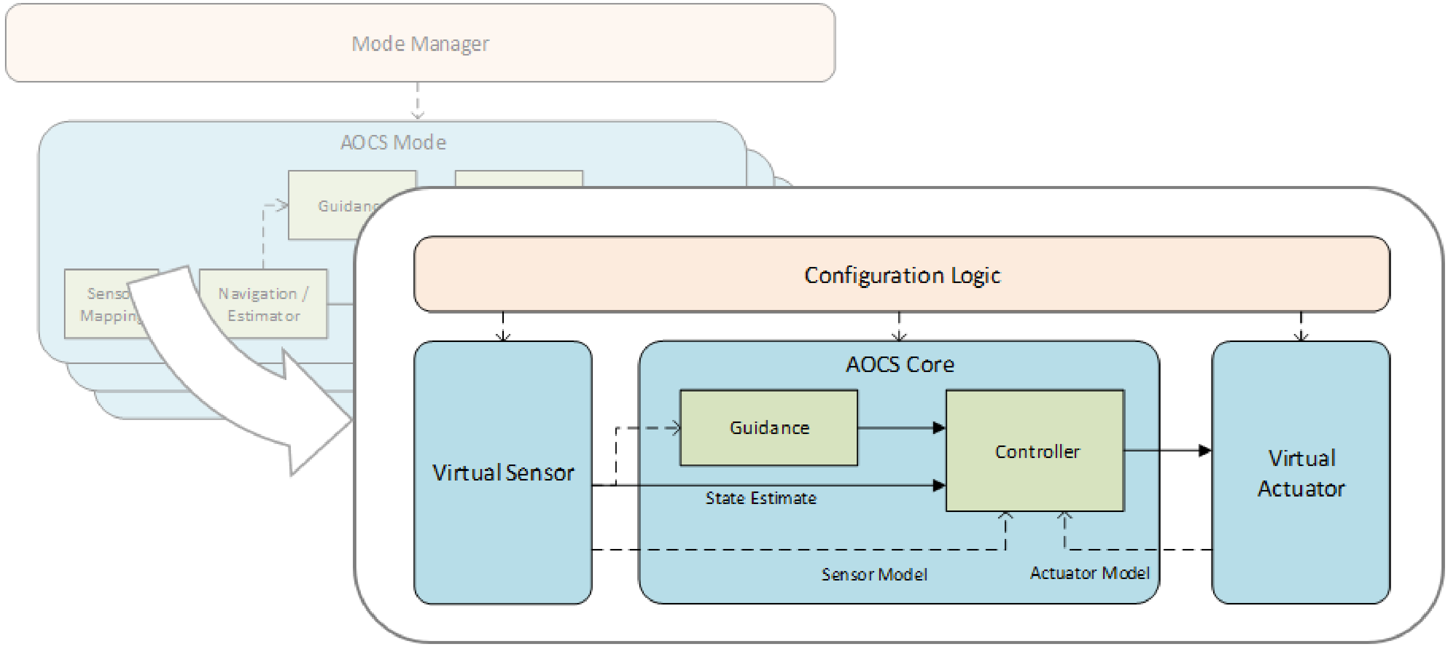
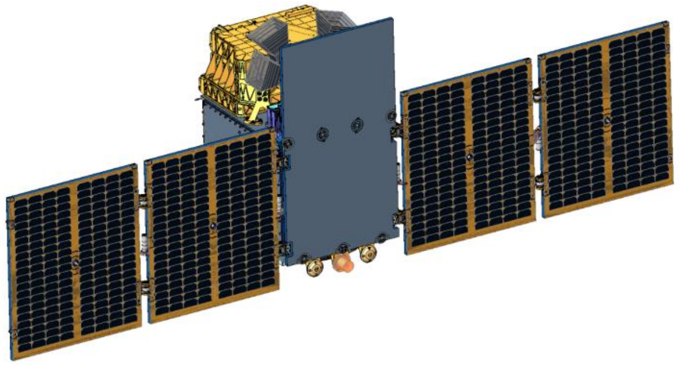
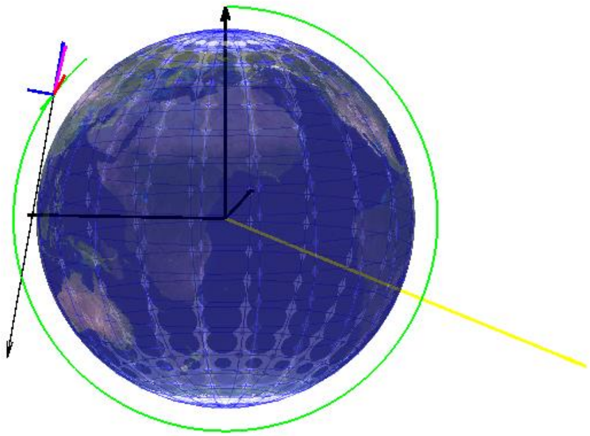
27-11-2024







Final Presentation

Virtual sensors and actuators for all-in-one mode AOCS

www.aeroespacial.sener

Final Presentation

Content

- AOCS-ONE project description
- Benchmark mission description (ALTIUS)
- Proposed single-mode AOCS architecture
- Design summary
- Test results and comparison with baseline
- Summary, conclusions, and recommendations

Project description

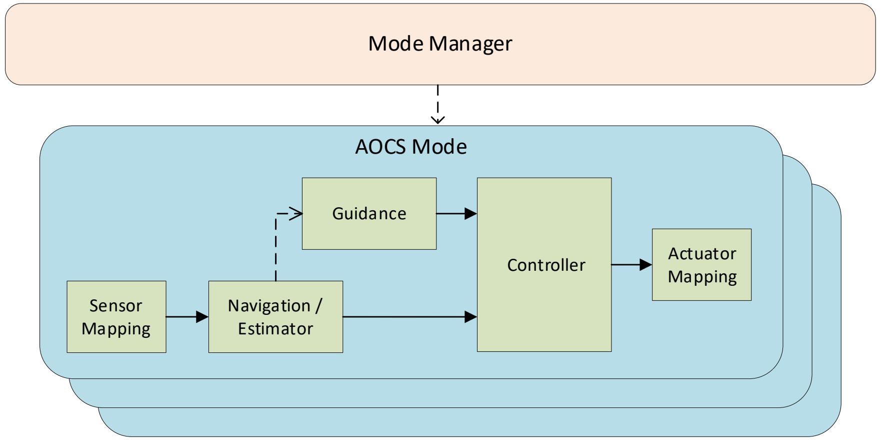
Motivation and aim of the activity

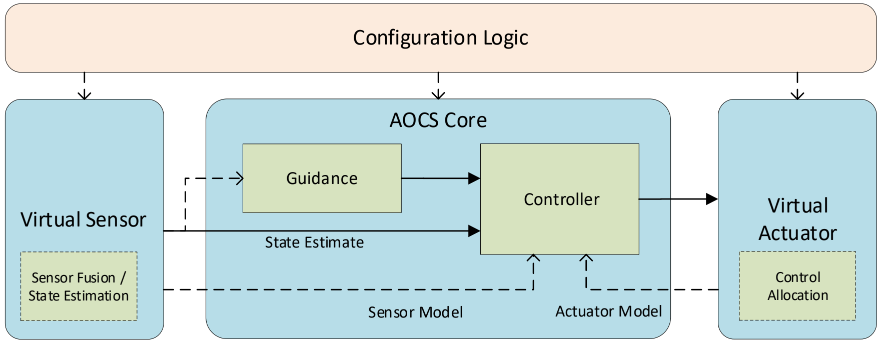
Background

- Conventional AOCS SW typically employs multiple modes applicable to different mission (sub-) phases to cover the full range of S/C functionality
- Often, these modes provide similar low-level functionality (suitable control of S/C attitude and rates, positions/velocities, etc.)
- This motivates the investigation of alternative AOCS architectures

Aim of the activity

- Exploit these similarities in low-level functionality to investigate and mature a novel AOCS architecture consisting of only a single mode.
- Different S/C configurations and tasks (e.g., rate damping, slews, course/fine pointing, science, or delta-V manoeuvres) are then to be accounted for using so-called virtual sensors (VS) and virtual actuator (VA) abstractions, as well as advanced control techniques allowing for adaptations to different mission phases and S/C configurations.





Project description

Expected benefits and challenges

Benefits	Description
Benefit 1	A decrease in verification and testing efforts due to the reduction of AOCS (sub-)modes and, hence, the reduction of test scenarios
Benefit 2	An enhanced HW modularity through the simplification of the replacement of one HW unit by another (i.e., the VS and VA abstract layers would only need to be parametrically adapted to different unit types and configuration)
Benefit 3	A potential reduction in the number of HW units thanks to a different redundancy approach through the VS and VA concept
Benefit 4	A simplified FDIR (less monitors, recovery actions, and mode dependencies)
Benefit 5	Simplified operations (less TCs, procedures, operational constraints)
Benefit 6	Easier re-use of AOCS SW components
Benefit 7	Increase of the spacecraft's autonomy

Challenge	Description	Potential solution
Challenge 1	**Modularity and encapsulation** • Used to break down complex requirements and functionalities into manageable pieces and reduce the cross-interaction between them. • Classically, this is (partly) realized using operational modes allowing for a practical abstraction and compartmentalization. • The different modes are isolated, allowing for mode-independent design and analysis, limiting the interaction to reasonably few well-defined mode switches. • This is different in a single-mode architecture where modularity and encapsulation are less pronounced at the level of system functionality.	Address the reduced modularity at the level of system functionality by an increased modularity and encapsulation at the technical functionality level (e.g., via VS and VA concepts which reduces the overall design complexity).
Challenge 2	**Task-specific functionality and performance** • Traditional modes aid tailoring the design to a specific task at hand. • Consequently, the performance is highly tuned for specific tasks.	• advanced control solutions • adaptive state estimation and advanced sensor fusion • optimized control allocation
Challenge 3	**Solution complexity** Traditional mode structures aid reducing the complexity of the technical AOCS solution, e.g., by enabling the use of relatively simple LTI filters and controllers.	Reduce complexity by avoiding modes and sub-modes while employing advanced but structured and modular solutions on a technical level

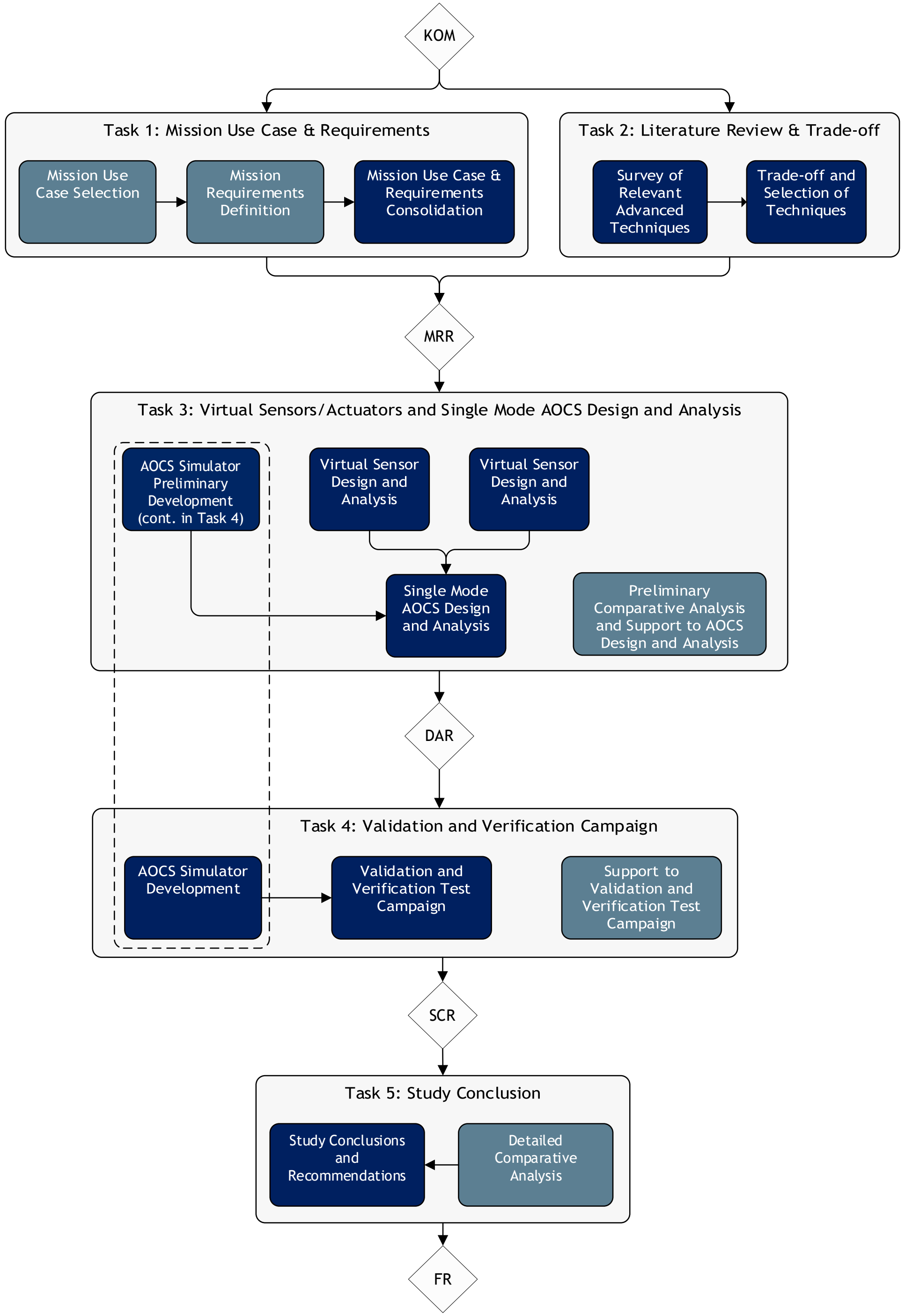
Project description

Study logic

The study is organized in 5 main tasks following a gradual development process:

- **Task 1:** Mission use case selection and requirements definition
- **Task 2:** Literature review and trade-off analysis of relevant advanced techniques
- **Task 3:** Virtual sensor/actuator and single mode AOCS design and analysis
- **Task 4:** Validation and verification campaign in functional simulator
- **Task 5:** Study conclusions and recommendations

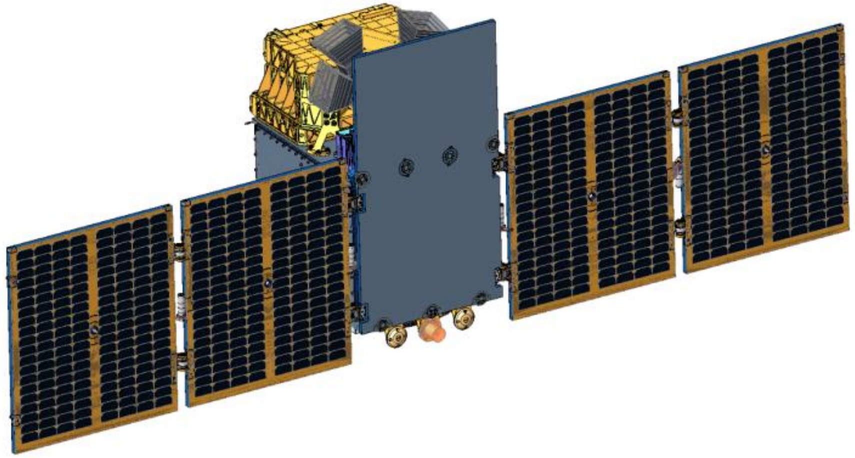
In the remainder of the presentation, we will present the outcome and obtained results

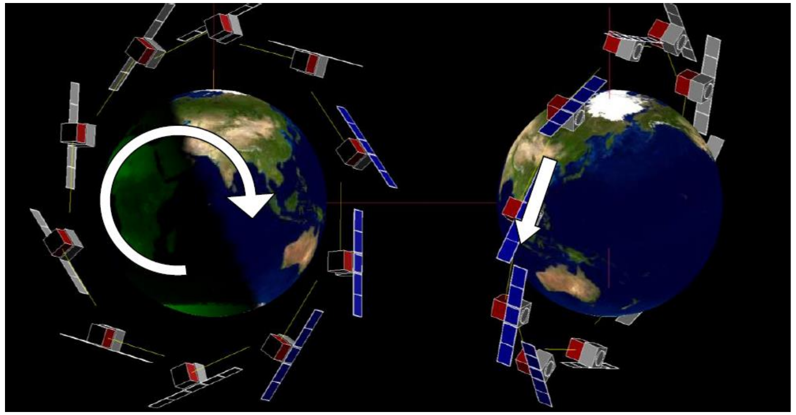


Benchmark mission

ALTIUS

- The mission use case is intended to provide context and a reference framework for the design and validation of the single mode AOCS.
- It should be sufficiently representative and challenging in terms of AOCS requirements, and sufficiently generic to demonstrate that the single mode AOCS can be applied EO missions.
- A trade-off was performed, and a selection was made from a series of accessible missions, namely: Proba-1, Proba-2, Proba-3, Proba-V, OKDSat, IOD, SAOCOM-CS, SBSS, and ALTIUS.
- ALTIUS (Atmospheric Limb Tracker for Investigation of the Upcoming Stratosphere) was the mission with the highest trade-off score according to the objectives:
    - The number of guidance modes and pointing profiles is high
    - The required agility and pointing precision is challenging
    - It is representative as a small-sat LEO Earth Observation mission
    - It hosts a wide range of sensors and actuators
    - The AOCS SW is fully accessible by Redwire and has already passed the acceptance review.
- ALTIUS is an ESA limb-sounding mission aiming at monitoring ozone concentrations in the atmosphere by means of a hyper-spectral 2D imager.
- ALTIUS is currently in phase C/D and the AOCS S/W passed the acceptance review. Despite not being spaceborne yet, its AOCS characteristics make up for this potential drawback.
- ALTIUS HW: 3x STR, 2x IMU, 2x GNSS, 3x MM, 4x TM, 4x RWL, 3x MT, 4x THR





Benchmark mission

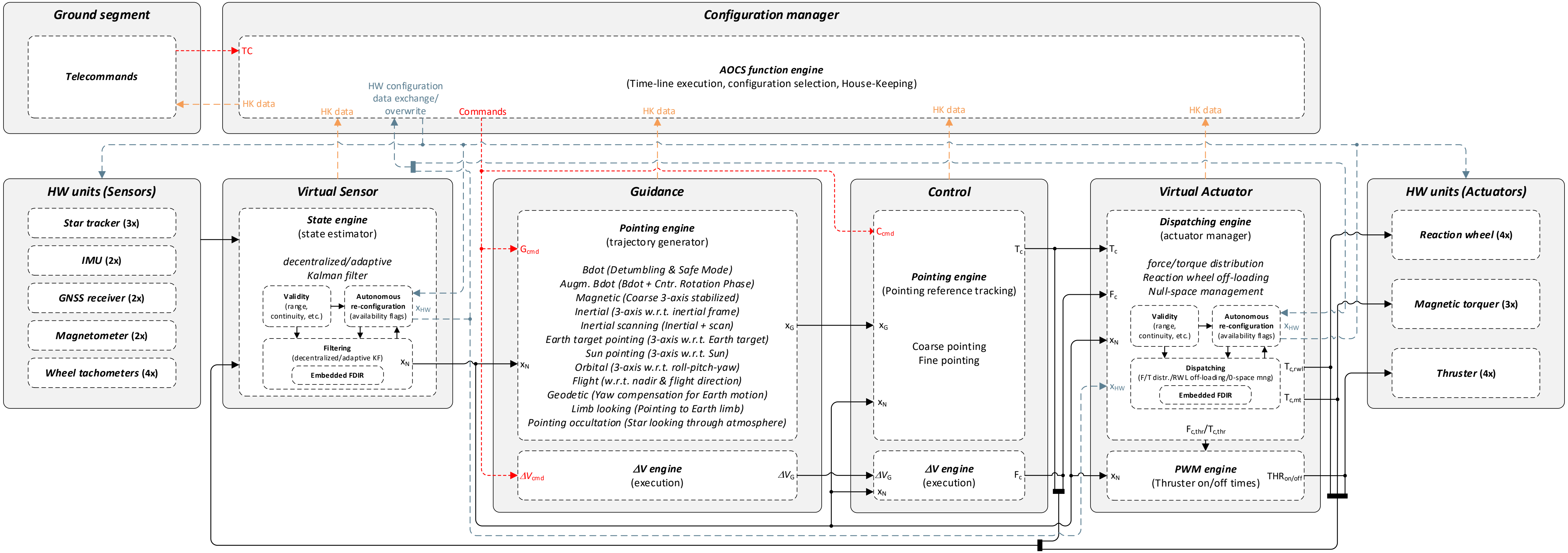
ALTIUS: Modes and hardware

AOCS mode	Sub-modes	MM	STR	IMU	GNSS	MT	RW	THR
Inertial mode	Fixed Scanning	X	X			X	X (3/4)	
Flight mode		X	X	X	X	X	X (3/4)	X
Geodetic mode	Yaw steering on/off	X	X		X	X	X (3/4)	
Earth target mode		X	X		X	X	X (3/4)	
Sunbathing mode		X	X		X	X	X (3/4)	
Limb looking mode	Backward/Left/ Optimal/Polar Tomography	X	X		X	X	X (3/4)	
Occultation mode	StandbyTracking	X	X		X	X	X (3/4)	

Proposed architecture

Proposed architecture

Virtual sensors and actuators for-all-in-one mode AOCS



Ground segment

Telecommands

Configuration manager

AOCS function engine
(Time-line execution, configuration selection, House-Keeping)

HW units (Sensors)

- ***Star tracker*** (3x)
- ***IMU*** (2x)
- ***GNSS receiver*** (2x)
- ***Magnetometer*** (2x)
- ***Wheel tachometers*** (4x)

Virtual Sensor

State engine
(state estimator)

decentralized/adaptive
Kalman filter

Validity (range, continuity, etc.)

Autonomous re-configuration (availability flags)

Filtering (decentralized/adaptive KF)

Embedded FDIR

Guidance

Pointing engine
(trajectory generator)

Bdot (Detumbling & Safe Mode)
Augm. Bdot (Bdot + Cntr. Rotation Phase)
Magnetic (Coarse 3-axis stabilized)
Inertial (3-axis w.r.t. inertial frame)
Inertial scanning (Inertial + scan)
Earth target pointing (3-axis w.r.t. Earth target)
Sun pointing (3-axis w.r.t. Sun)
Orbital (3-axis w.r.t. roll-pitch-yaw)
Flight (w.r.t. nadir & flight direction)
Geodetic (Yaw compensation for Earth motion)
Limb looking (Pointing to Earth limb)
Pointing occultation (Star looking through atmosphere)

ΔV engine
(execution)

Control

Pointing engine
(Pointing reference tracking)

Coarse pointing
Fine pointing

ΔV engine
(execution)

Virtual Actuator

Dispatching engine
(actuator manager)

force/torque distribution
Reaction wheel off-loading
Null-space management

Validity (range, continuity, etc.)

Autonomous re-configuration (availability flags)

Dispatching (F/T distr./RWL off-loading/0-space mng)

Embedded FDIR

PWM engine
(Thruster on/off times)

HW units (Actuators)

- ***Reaction wheel*** (4x)
- ***Magnetic torquer*** (3x)
- ***Thruster*** (4x)

Proposed architecture

Proposed architecture: Safe mode vs. safe configuration

ALTIUS Safe modes

The ALTIUS AOCS includes various safe modes, all of which are (quasi) spin-stabilized:

- **B-dot/Augmented B-dot mode:** Uses MTs and RWLs running at constant speed.
- **Magnetic mode:** Uses MM measurements, MT actuation with the RWLs held at a constant speed and the spacecraft position provided either by two-line elements (TLE) of GPS.

For a single mode AOCS architecture, these configurations are not trivial to realize without introducing modes (although not impossible).

Main assumption -> The AOCS-ONE design is 3-axis stabilized under all conditions

AOCS-ONE Safe configuration

Instead of designing a separate **safe mode** (which would be violating the single mode AOCS notion), we covered the safety aspect by means of a **safe configuration**

- The safe mode functionality is essentially integrated in the “nominal” AOCS
- The AOCS just reconfigures to using a subset of the available hardware

Selected techniques

Overview

Virtual sensors and actuators for-all-in-one mode AOCS

Navigation

❖ Multi-sensor no-reset Federated Kalman Filter (FKF)

- Decentralized and decomposed KF
- Weighted snapshot fusion
- Default for local filters: EKF (Extended Kalman Filter)
- Innovation-based adaptive estimation (IAE) techniques: covariance matching and innovation chi-square testing (ICST)

Control

❖ H_∞-control, Youla-switching

- Design H_∞-controllers for tasks (slew, pointing, safe)
- Youla-based controller re-parameterization
- Optimal performance for each task
- (Smooth) switching between controllers with stability guarantees

Control Allocation

❖ Convex Optimization

- Quadratic Programming with Polytopic Constraints
- Developments to formulate hybridization of actuators and frequencies
- Interior Point Method (IPM), customized solver for the application
- Based on heritage from Sener Optimization ToolBox (SOTB)

Guidance

❖ Sequential Convex Programming

- Lossless convexifications for pointing constraints
- Geometric computation of targets in inertial frame
- IPM for convex iterations
- Based on heritage from SOTB

Virtual Sensor (VS)

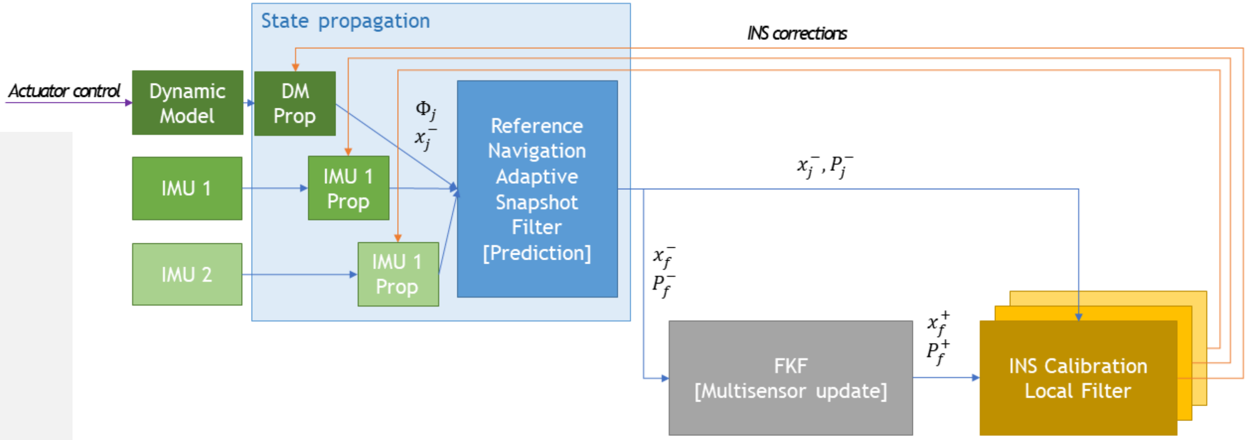
Design summary

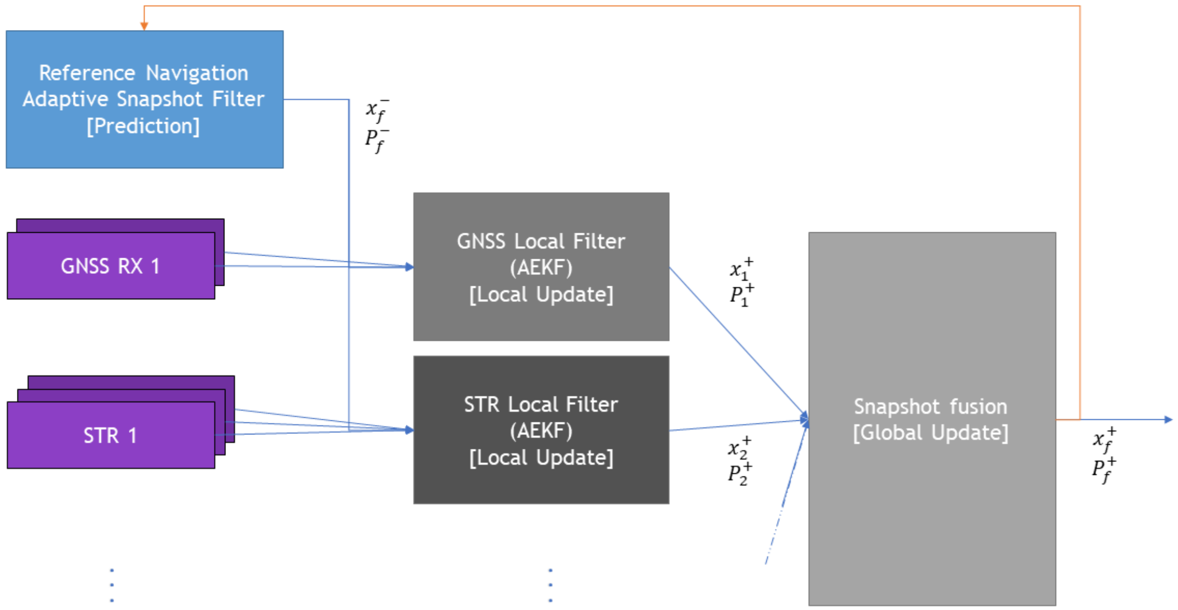
The Virtual Sensor (VS) acts as an **abstraction layer** for the sensing and navigation functions, with **autonomous reconfiguration** capabilities and **adaptability** to different scenarios, it provides a **uniform interface** regardless of its internal configuration.

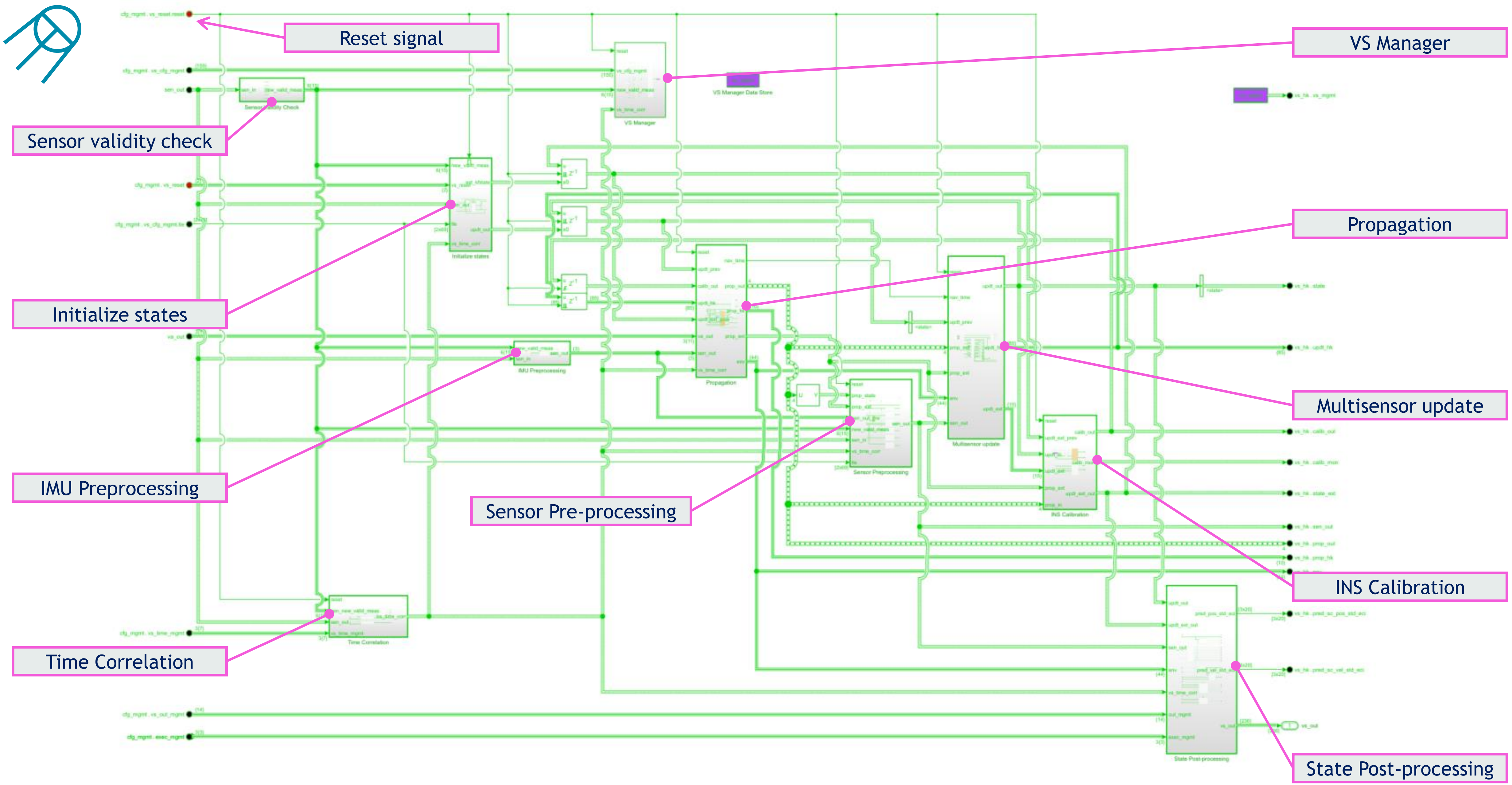
The VS implements a **decentralized multi-sensor hybridization based on a federated (FKF) no-reset** approach with decomposed prediction and update steps. The global update is performed though a **snapshot fusion** of the EKF local updates. The **reference navigation** is computed through the fusion of three independent propagation models with **adaptative process noise** and **calibrated** with the results of the updated state. At each level there is **embedded FDIR** for improved robustness and avoiding fault propagation downstream the GNC system.

The VS estimates the **6-DoF state**, and an extended state used to obtain additional parameters and calibrate the dynamic model. It also provides **ancillary services** such as time-keeping, event prediction, etc.

Virtual sensors and actuators for-all-in-one mode AOCS


State propagation
INS corrections
Actuator control
Dynamic Model
DM Prop
IMU 1
IMU 1 Prop
IMU 2
IMU 1 Prop
Φ_j
x_j^-
Reference Navigation Adaptive Snapshot Filter [Prediction]
x_j^-, P_j^-
x_f^-
P_f^-
FKF [Multisensor update]
x_f^+
P_f^+
INS Calibration Local Filter


Reference Navigation Adaptive Snapshot Filter [Prediction]
x_f^-
P_f^-
GNSS RX 1
STR 1
GNSS Local Filter (AEKF) [Local Update]
STR Local Filter (AEKF) [Local Update]
x_1^+
P_1^+
x_2^+
P_2^+
Snapshot fusion [Global Update]
x_f^+
P_f^+



Reset signal

VS Manager

Sensor validity check

Propagation

Initialize states

Multisensor update

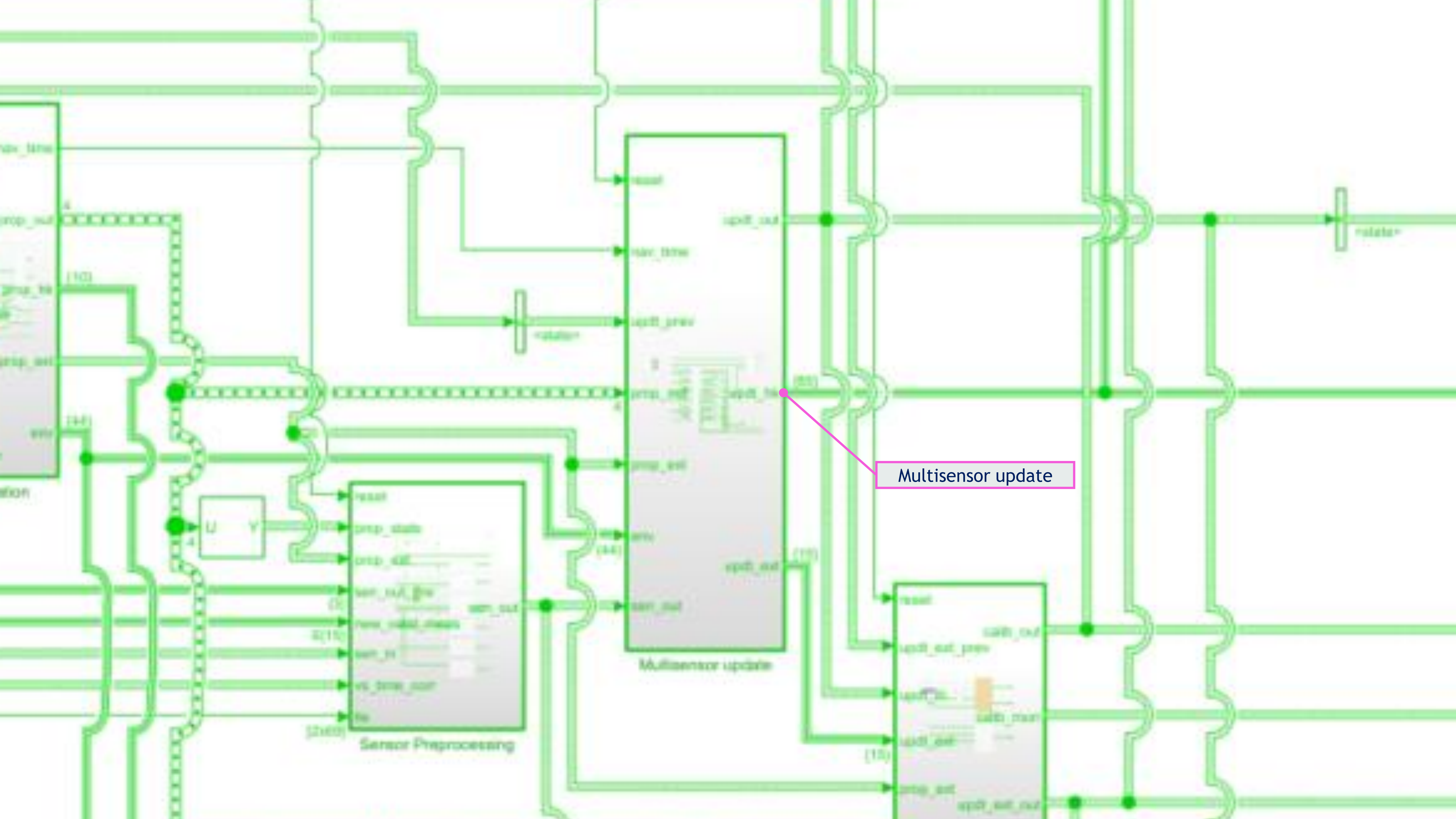
IMU Preprocessing

Sensor Pre-processing

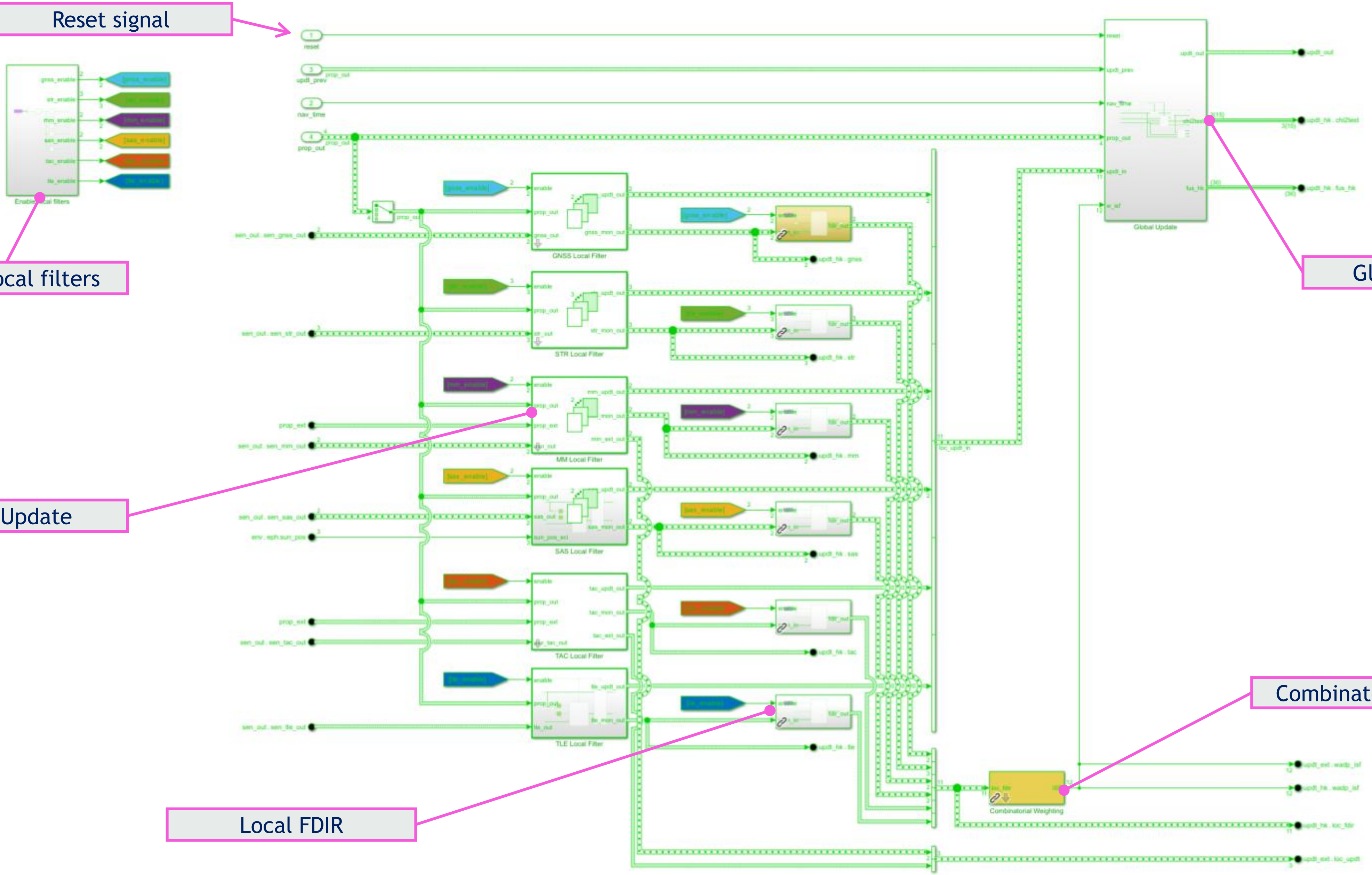
INS Calibration

Time Correlation

State Post-processing



Multisensor update



Reset signal

Enable local filters

Local Update

Local FDIR

Global Update

Combinatorial Weighting

Virtual Sensor (VS)

Technique selection: Snapshot weighted fusion

Previous to the global update, the outputs from the local updates are checked through a **chi-square test**, which obtains the test statistic q:

$$q = \Delta x^T (P^{-1}) \Delta x \sim \chi^2(n)$$

Eq. 121

The test statistic is fuzzified to obtain the probability value β. In the multi sensor framework these probabilities can be combined to obtain the **probability space of the fusion**, where each λ is computed by multiplying the validation probabilities β of the sensors fused for each combination.

$$\lambda_G = \prod_i^G \beta_i$$

Eq. 122

The **weight of each combination α_G** is computed using the inclusion-exclusion principle. Finally, the weight of the propagation-only update, meaning that the rest of updates are not valid or have failed, is computed by subtracting the rest of weights to 1.

$$\alpha_G = \bigcup_i^G \lambda_i$$
$$\alpha_P = 1 - \sum_k^N \alpha_k$$

Eq. 123

Using the snapshot fusion, the local update can be weighted using **the information sharing factors κj**. Combining the weights for each update the corresponding ISF would be the sum of weights of each group that includes that update divided by the number of updates in said group:

$$\kappa_j = \sum_G^{k \in G} \frac{\alpha_G}{N_G}$$

Eq. 124

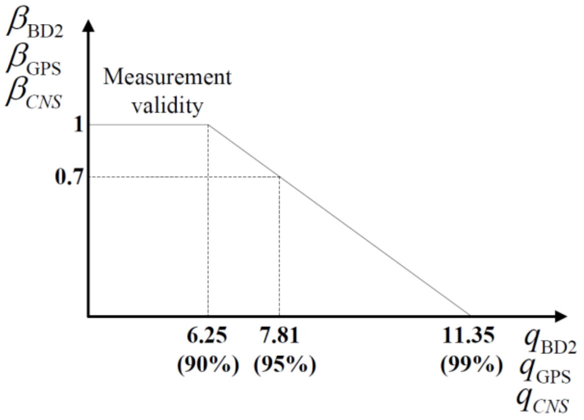
Finally, the outputs from the local updates are combined through **a weighted snapshot** or least-squares algorithm [RD114] using the ISF. The resulting fused state and covariance matrix are then:

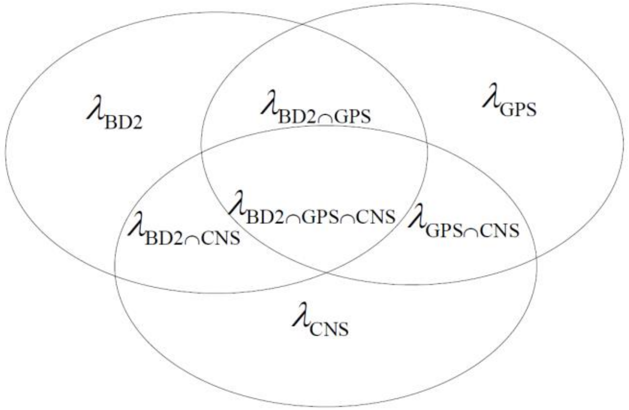
$$(P_k^+)^{-1} = \sum \left(\kappa_j^{-1} P_{j,k}^+\right)^{-1}$$

Eq. 109

$$\delta x_k^+ = P_k^+ \sum \left(\left(\kappa_j^{-1} P_{j,k}^+\right)^{-1} \delta x_{j,k}^+ \right)$$

Eq. 110





Virtual Sensor (VS)

Technique selection: Q-adaptation

The adaptation of the process noise covariance matrix Q is performed following the covariance matching technique. The innovation covariance can be estimated though a window averaging of the innovation as [RD49]

$$\hat{S} = \frac{1}{N}\sum r r^T \qquad \text{Eq. 83}$$

Then the process noise covariance can be adapted following [RD181]

$$\hat{Q} = K\hat{S}K^T \qquad \text{Eq. 84}$$

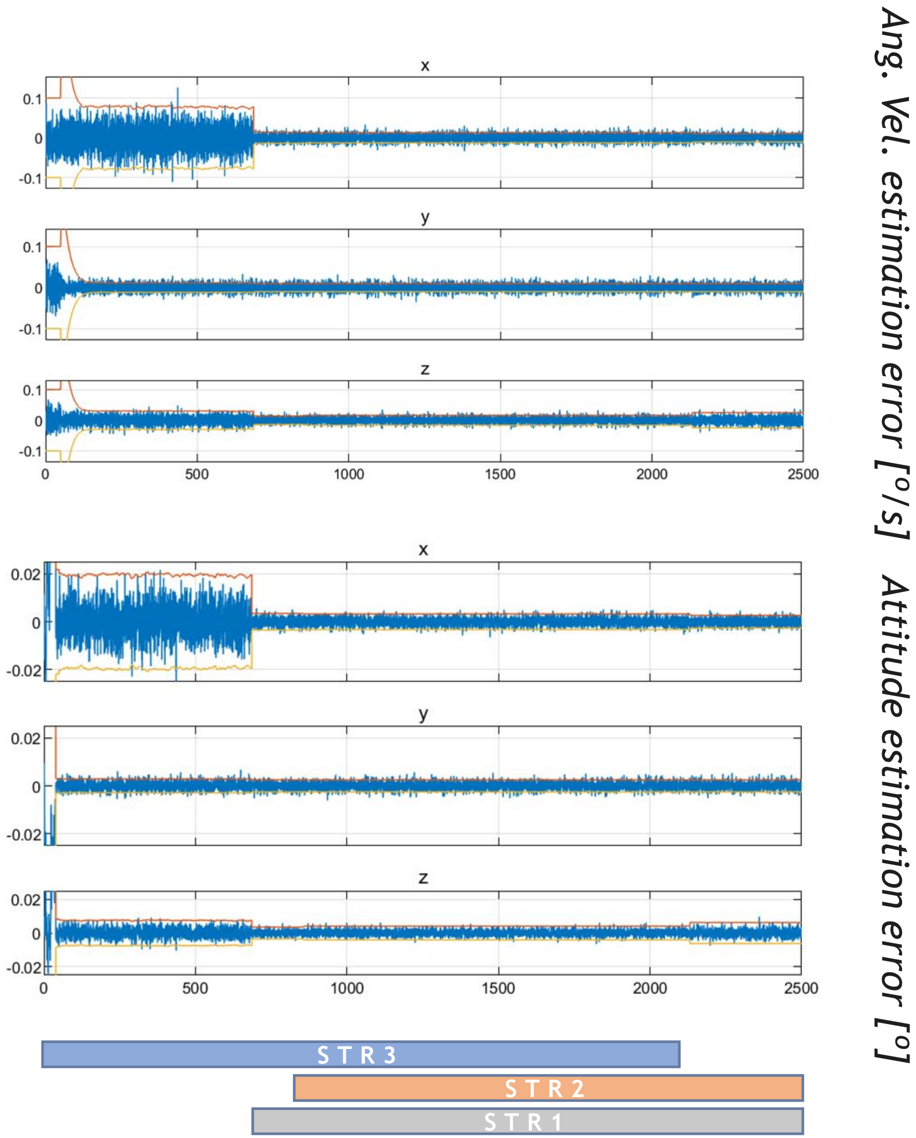
Where K is the Kalman gain. Combining both equations one can derive that:

$$\hat{Q} = K\frac{1}{N}\left(\sum r\, r^T\right)K^T = \frac{1}{N}\sum K r r^T K^T = \frac{1}{N}\sum Kr\,(Kr)^T \qquad \text{Eq. 85}$$

Then, by definition $\delta x_k^+ = \delta x_k^- + K_k r_k$, which yields:

$$\widehat{Q_j} = \frac{1}{N}\sum\left(\delta x^+ - \delta x_j^-\right)\left(\delta x^+ - \delta x_j^-\right)^T \qquad \text{Eq. 86}$$

Which can be used to estimate Q_j independently for each propagation model using the fused *a posteriori* state δx^+ and the local *a priori* propagated state δx_j^- by each propagation model.



Pros & Cons

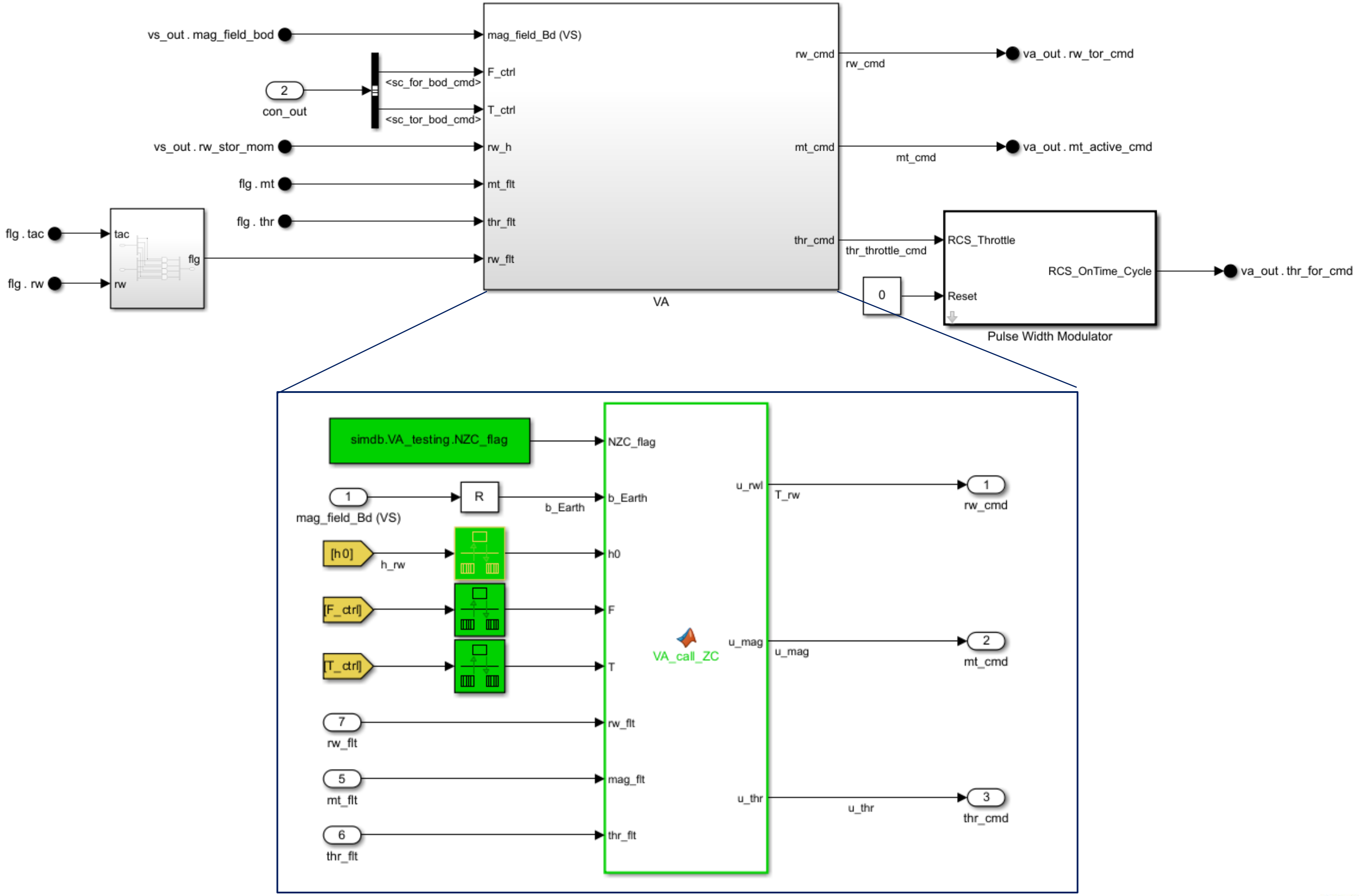
Virtual Sensor (VS)

- ✓ Modular and scalable design
- ✓ High level of autonomy
- ✓ Less reliant on tuning (lower tuning effort)
- ✓ Adaptive filter for wide range of scenarios/operational points
- ✓ No mode dependency
- ✓ Abstraction layer with uniform interface for downstream subsystems
- ✓ Generality of solution for reusable design
- ✓ Lower V&V effort in successive usages of the subsystem
- ✓ Embedded FDIR
- ✓ Robustness to faults

- ❖ Higher computational load
- ❖ Higher complexity of algorithms
- ❖ Lower performance wrt a centralized multi-sensor approach
- ❖ Higher operational range can incur in higher V&V effort of the subsystem

Virtual actuator

VA block



Virtual actuator

Technique selection

Trade-off

High-level perspective:

- Abstraction layer
- Actuation provider
- Actuator manager

Specific:

- Generic.
- Implement FDIR functionalities.
- Autonomous and independent.
- Flexible.
- Real time.
- V&V compatibility.
- Extra functionalities.

Convex Optimization

$$\min_{U\in\mathbb{R}^n,\, h\in\mathbb{R}^{n_R}} \quad \frac{1}{2}U^{\mathrm{T}}RU + r^{\mathrm{T}}U + \frac{1}{2}\left(h-h_{\mathrm{ref}}\right)^{\mathrm{T}}W\left(h-h_{\mathrm{ref}}\right) + \frac{1}{2}\left(f-BU\right)^{\mathrm{T}}L\left(f-BU\right) \qquad (10a)$$

$$\text{s.t.} \quad u_i^{\min} \le u_i \le u_i^{\max}, \quad i\in\{1,\cdots,n\}, \qquad (10b)$$

$$h_j = h_{0j} + \Delta t\cdot u_j, \quad j\in\{1,\cdots,n_R\}, \qquad (10c)$$

$$h_j^{\min} \le h_j \le h_j^{\max}, \quad j\in\{1,\cdots,n_R\}. \qquad (10d)$$

f	Command (Force and Torque)
u_i	Actuation variables
h_i	Stored momentum in RWLs
h_{ref}	Reference preferred momentum
B	Force and torque control matrix

R, W, L	Weighting matrices
h_{i_0}	Current stored momentum in RWLs
Δt	Control time step
u_i^{max}, u_i^{min}	Control constraints
h_i^{max}, h_i^{min}	RWLs momentum constraints

Achievable objectives with the selected proposal :

1. **Optimal allocation of command** while using the cheapest actuator available exploiting null space.
2. **Error minimization** in case commanded torque/force is not feasible
3. **Autonomous continuous RWLs desaturation.**
4. **Autonomous management of failed actuators** to provide commands with minimum error.
5. **Hybridization.**

Virtual actuator

Generic formulation

<u>Developments for computational load reduction:</u>

- Stored momentum variables removed using equality constraint.
- Momentum and torque saturation bounds merged into a single (RWLs-wise) constraint.
- Actuation variables normalized to avoid numerical issues and ease the tuning process.
- Cost function grouped into quadratic and linear terms to be updated with inputs.
- Constant parts of the cost function removed.
- Separation in torque and force contribution to ease tuning

```
$\boldsymbol{VA\ inputs}: F, T, h_0, b_{Earth}, i_{fail}$
$\boldsymbol{Magnetic\ torquer\ effect\ update\ with\ magnetic\ field\ in\ body\ frame\ (b_E)}$:
$B(:, i_{MAG}) = B_{MAG}(b_E);$
$\boldsymbol{Failed\ units\ deactivation}$:
$B(:, i_{fail}) = 0;$
$A(:, i_{fail}) = 0;$
$h_0(i_{fail} \le \boldsymbol{n_R}) = 0;$
$\boldsymbol{Thrusters\ deactivation\ out\ of\ \Delta V\ (optional)}$:
$if\ F = 0$
    $B(:, i_{THR}) = 0;$
$end$
$\boldsymbol{Saturation\ to\ avoid\ unfeasibilities\ (protection)}$:
$h_0(|h_0| > h_{max}) = \frac{h_0(|h_0| > h_{max})}{|h_0(|h_0| > h_{max})|} h_{max};$
$\boldsymbol{Solve\ optimization\ problem}$
$\boldsymbol{VA\ Outputs}: u^* = U_{max}\bar{u}^*$
```

General VA problem formulation:

$$\min_{\bar{u} \in R^n} \quad \frac{1}{2}\bar{u}^T H \bar{u} + c^T \bar{u};$$

$$s.t. \quad -1 \le \bar{u}_i \le 1; \qquad (MT);$$

$$0 \le \bar{u}_i \le 1; \qquad (THR);$$

$$max\left(\frac{h_{i_{min}} - h_{i_0}}{\Delta t u_{i_{max}}}, -1\right) \le \bar{u}_i \le min\left(\frac{h_{i_{max}} - h_{i_0}}{\Delta t u_{i_{max}}}, 1\right); \qquad (RWLs);$$

with:

$$H = R + U_{max}^T B^T Q B U_{max} + U_{max}^T A^T G A U_{max} + U_{max}^T \Delta T^T \begin{bmatrix} W & 0 \\ 0 & 0 \end{bmatrix} \Delta T U_{max}$$

$$c^T = -T^T Q B U_{max} - F G A U_{max} + \begin{bmatrix} h_0^T W & 0 \\ 0 & 0 \end{bmatrix} \Delta T U_{max};$$

$$\Delta T = \begin{bmatrix} \begin{bmatrix} \Delta t & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Delta t \end{bmatrix}_{4\times4} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{n\times n}$$

Virtual actuator

VA tailoring for ALTIUS mission

ALTIUS tailoring adds to the generic formulation:

- Dual frequency adaptation for thrusters (1 Hz), magnetic torquers (1 Hz), and reaction wheels (4 Hz). VA works at 4 Hz switching between two different formulations depending on the cycle.
- Reaction wheels zero-crossing minimization encouraged by use of non-soft cost and preferred working region.
- Actual torque-momentum constraint envelope.

$$\min_{\bar{U}, \in \mathbb{R}^{n}, \gamma \in \mathbb{R}^{n_R}} \quad \frac{1}{2}\bar{U}^{\mathsf{T}} H \bar{U} + c^{\mathsf{T}} \bar{U} + \frac{1}{2}\gamma^{\mathsf{T}} D \gamma + d^{\mathsf{T}} \gamma \tag{19a}$$

$$\text{s.t.} \quad g_{-}(h_{0j}) \leq \bar{u}_j \leq g_{+}(h_{0j}), \quad j \in \{1, \cdots, n_R\}, \tag{19b}$$

$$-1 \leq \bar{u}_i \leq 1, \quad i \in \{n_R + 1, \cdots, n_R + n_M\}, \tag{19c}$$

$$0 \leq \bar{u}_i \leq 1, \quad i \in \{n_R + n_M + 1, \cdots, n_R + n_M + n_T\}, \tag{19d}$$

$$\gamma_j \geq 0, \qquad j \in \{1, \cdots, n_R\}, \tag{19e}$$

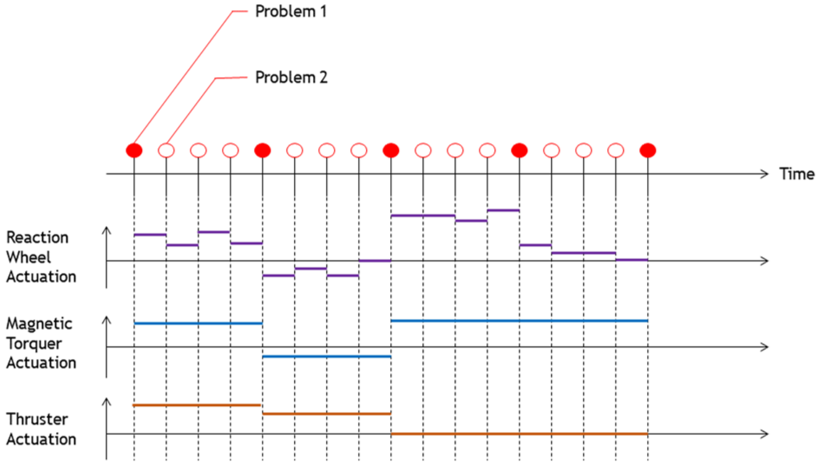
$$-\frac{h_{refj}}{|h_{refj}|}\Delta t \cdot u_j^{\max} \bar{u}_j - \gamma_j \leq \frac{h_{refj}}{|h_{refj}|} h_{0j} - |h_{zc}|, \quad j \in \{1, \cdots, n_R\}, \tag{19f}$$

$$g_{-}(h_{0j}) = \min\left(\max\left(\frac{h_j^{\min} - h_{0j}}{\Delta t \cdot u_j^{\max}}, -1, -1 + m(h_{0j} - h_m)\right), 0\right), \quad j \in \{1, \cdots, n_R\}, \tag{19g}$$

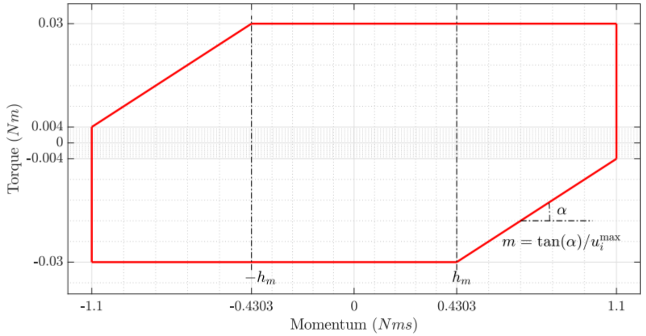
$$g_{+}(h_{0j}) = \max\left(\min\left(\frac{h_i^{\max} - h_{0j}}{\Delta t \cdot u_j^{\max}}, 1, 1 - m(h_{0j} + h_m)\right), 0\right), \quad j \in \{1, \cdots, n_R\}. \tag{19h}$$

$$H = R + \bar{B}^{\mathsf{T}} L \bar{B} + \operatorname{diag}\left(\left(\Delta t \cdot u_1^{\max}\right)^2 w_1, \cdots, \left(\Delta t \cdot u_{n_R}^{\max}\right)^2 w_{n_R}, 0\right) \qquad c = r - f^{\mathsf{T}} L \bar{B} + \begin{bmatrix} \Delta t \cdot u_1^{\max} \cdot w_1 \left(h_{01} - h_{ref1}\right) \\ \vdots \\ \Delta t \cdot u_{n_R}^{\max} \cdot w_{n_R} \left(h_{0n_R} - h_{refn_R}\right) \\ 0 \end{bmatrix}.$$

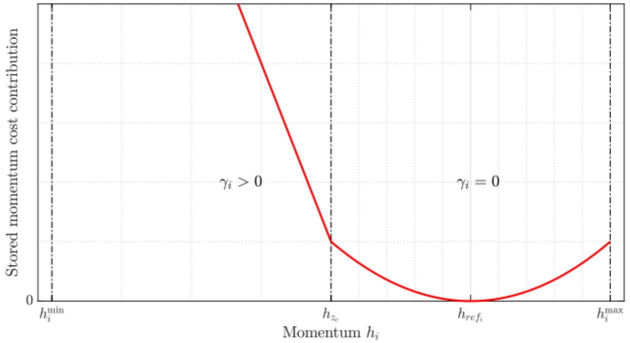
Sketch of dual frequency adaptation, cost function also modified for synchronization:



Implementation of the actual torque-momentum constraint:



Implementation of soft constraint for zero crossing minimization:



Pros & Cons

Virtual actuator (VA)

- Autonomy
- Hybridization and control allocation
- Flexibility
- Optimality (performance)
- Actuator constraints consideration
- Full envelope exploitation (including torque-momentum relation)
- Null space management
- Continuous desaturation
- No mode dependency (but adaptable to modes)
- Fault robustness
- Error minimization for unrealizable commands
- Reusability and ease of use *(already being used in other SENER projects)*
- Scalability and extendibility
- V&V reduced thanks to reusability
- Modularity
- AOCS oversize reduction

- Complexity of (first) design
- Needing of dedicated SW
- Convexity limitations (not with respect to classical approach)
- Computation cost (real-time, but higher than classical approach)

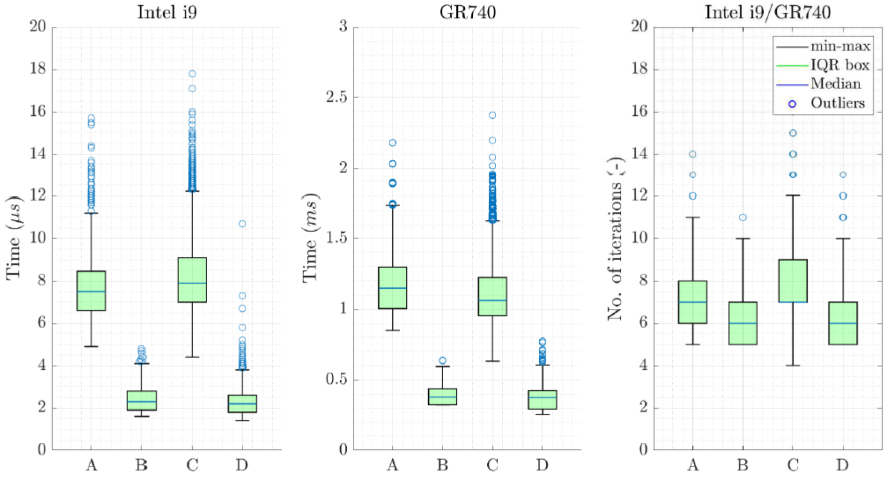


Fig. 13 Computational performance for the ALTIUS benchmark problem obtained with the customized solver: (A) nominal 1 Hz cycle, (B) nominal 4 Hz cycle, (C) degraded 1 Hz cycle, and (D) degraded 4 Hz cycle

Guidance

Guidance proposed architecture

VS inputs:

- Propagation of spacecraft (and target) position and velocity
- Current spacecraft, Sun, and RWs states

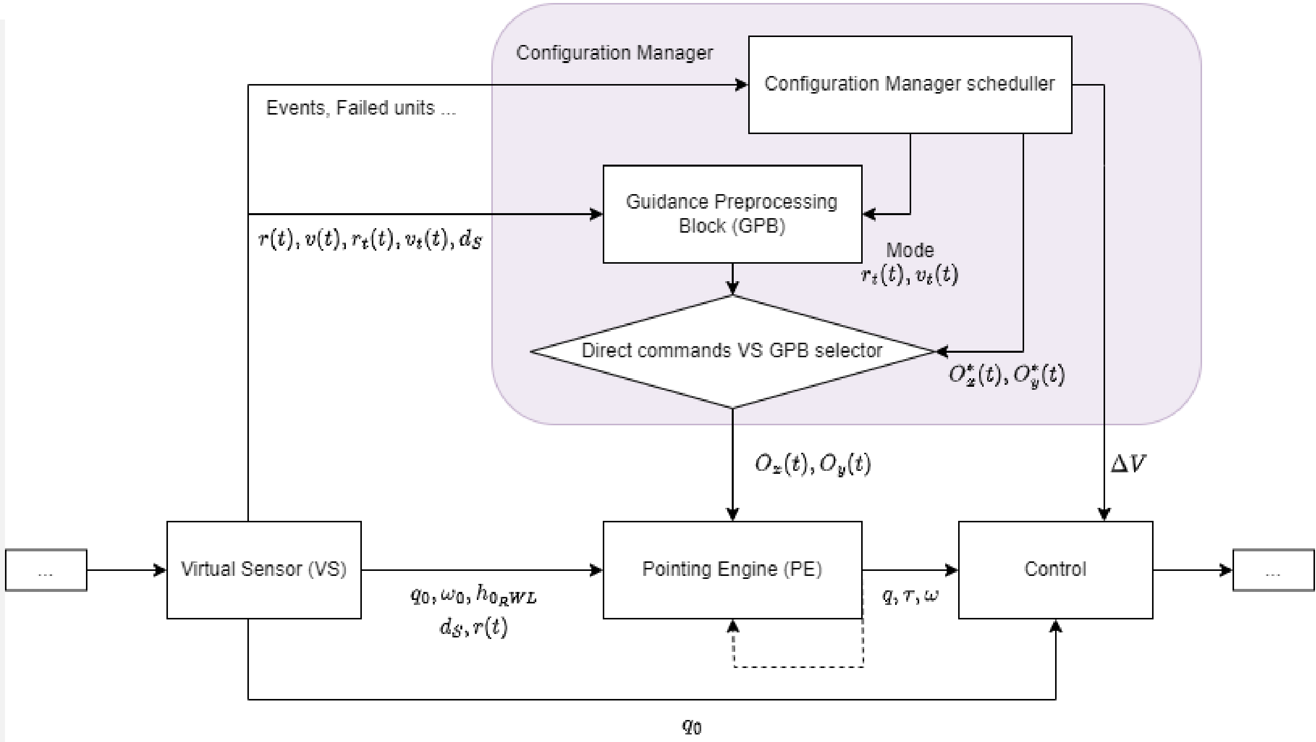
Configuration Manager (CM) tasks:

- Select target (scheduler/ground command) or send direct-inertial/Earth target
- Guidance Preprocessing Block (GPB): Computes autonomously target for PE

Pointing Engine (PE) tasks:

- Based on onboard optimization
- Generates smooth guidance feedforward profiles in terms of quaternion, velocity, and torque
- Generation (optimization) triggered by configuration manager (e.g., target change) and/or execution of previous profile
- Continuous smooth profiles (avoid control jumps) accounting for system and mission constraints
- Target agnosticism
- Acquisition (slew) + tracking merged

Virtual sensors and actuators for-all-in-one mode AOCS



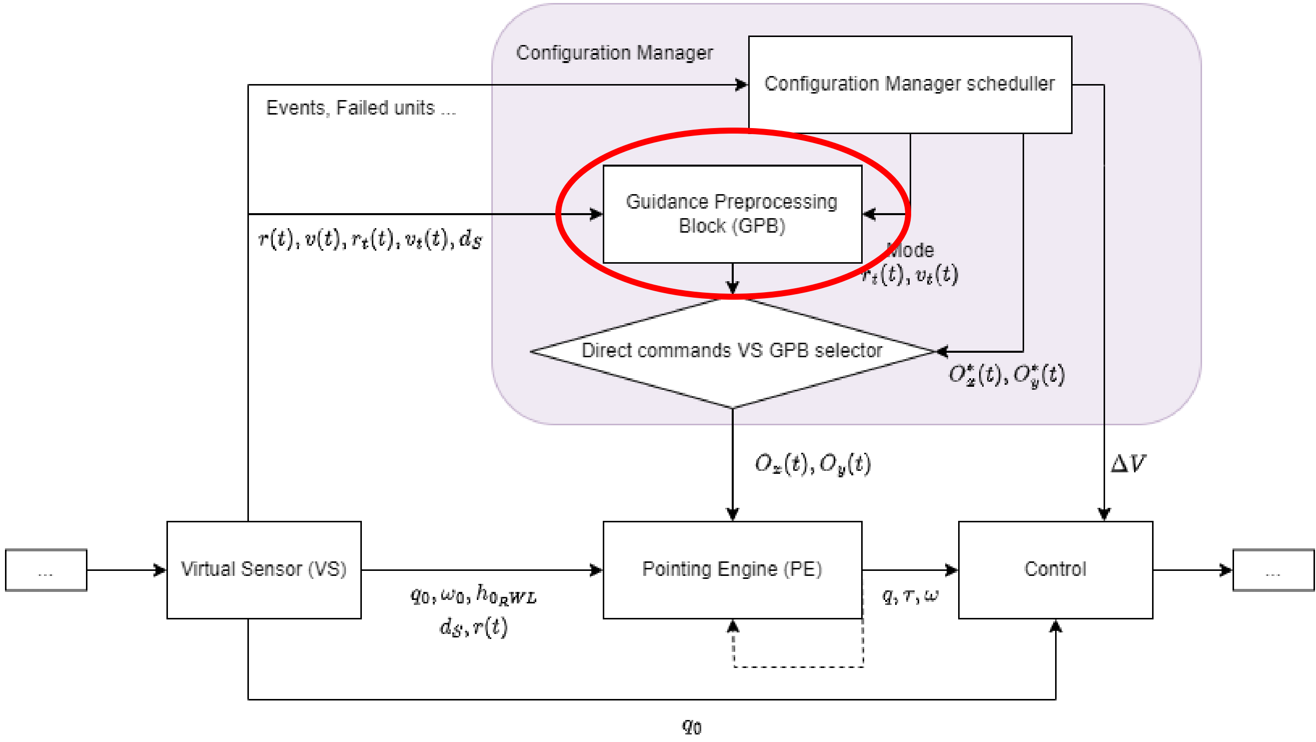
- Guidance of the GNC is reduced to the PE
- The GPB collects the mission targets, becoming the mission-dependent block that interacts with the all-in-one AOCS.
- The GPB can be updated during mission without changes on the GNC.

Guidance

Guidance Preprocessing Block

- Generates two target directions $(O_x(t), O_y(t))$ in inertial frame (original guidance frames removed)
    - Not necessarily continuous
    - Not necessarily feasible
- Receives required information from VS (current implementation) / Ground (potential enhancement)
- Can be updated without affecting the GNC
- For the same platform, this is the only block that need to be changed for different mission profiles

Virtual sensors and actuators for-all-in-one mode AOCS



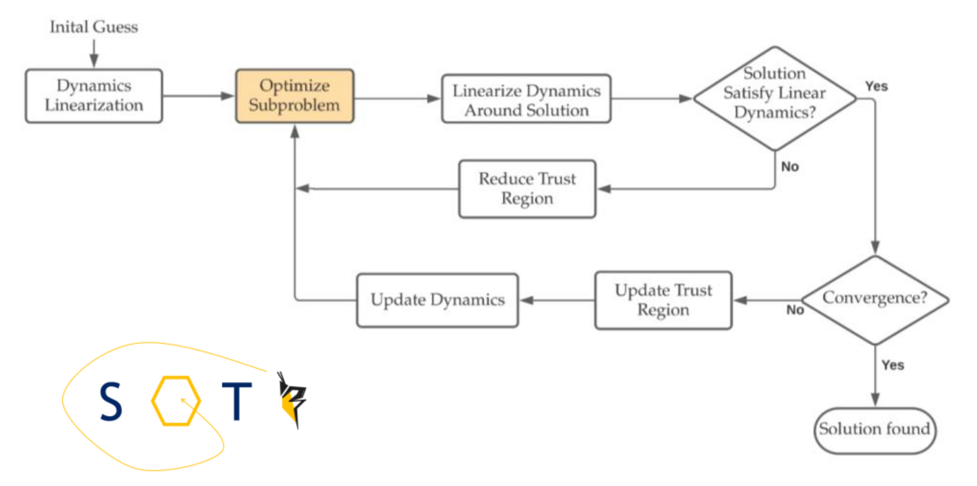
Guidance strategy

PE tailoring for ALTIUS mission

Virtual sensors and actuators for-all-in-one mode AOCS

Note: This tailoring is yet generic and can be used in many other missions

Sequential Convex Programming (SCP) + lossless convexification on pointing constraints and objectives



$$\min_{q_i,\omega_i,\tau_i} \frac{1}{2} q_N^T Q_N q_N + d^T\gamma + \sum_{i=0}^{N-1} \frac{1}{2}\tau_i^T R \tau_i + \frac{1}{2} q_i^T Q_i q_i\,;$$

$$s.t.\ \dot{q}_i = \frac{1}{2}\Omega(\omega_i) q_i;$$

$$I_b \dot{\omega}_i = \tau_i;$$

$$\frac{1}{2} q_i^T S_{ij} q_i \le s_i + \gamma_j;$$

$$A_\omega \omega_i \le b_\omega + A_\omega \omega_v\left(\overline{\omega_0}, \overline{h_0}_{RWL}\right);$$

$$A_\tau \tau_i \le b_\tau;$$

$$\left[q_0, \omega_0, h_{0\,RWL}\right]^T = \left[\overline{q_0}, \overline{\omega_0}, \overline{h_0}_{RWL}\right]^T;$$

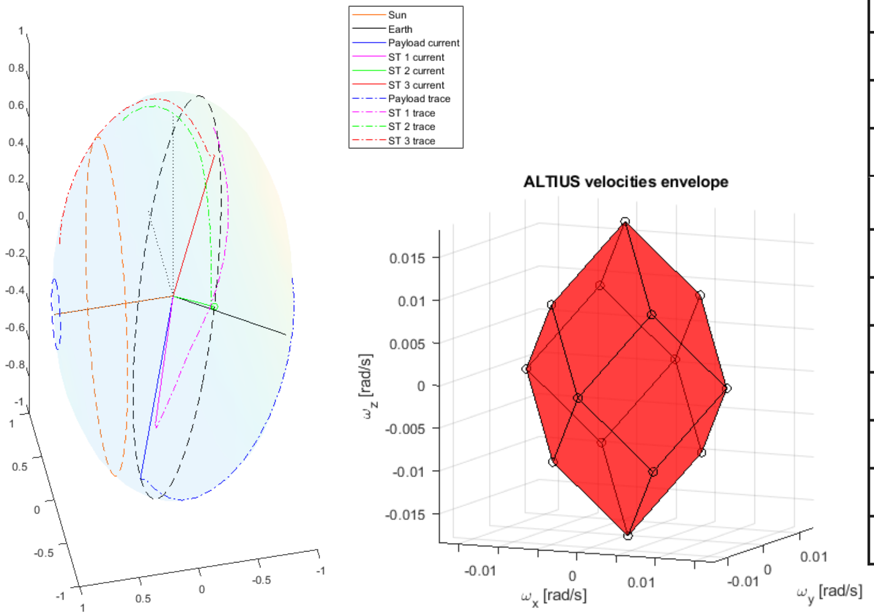
$$\omega_v\left(\overline{\omega_0}, \overline{h_0}_{RWL}\right) = I_b^{-1} B_{RWL} + \overline{\omega_0}.$$

Unique non-convexity

q	Quaternion of body frame w.r.t. inertial frame
ω	Spacecraft angular velocity in body frame
τ	Control torque
I_b	Inertia matrix
h_{RWL}	RW stored momentum
$\overline{q_0}, \overline{\omega_0}, \overline{h_0}_{RW}$	Initial conditions
i	Stage in the horizon
N	Horizon length
Q, W, R	Weighting matrices
$S\,, s$	Matrices for pointing constraints
A, b	RW actuation envelope
$T_{MAG}(q_i, t)$	Magnetic torquer actuation envelope
T_{THR}	Thruster actuation envelope

Achievable objectives with the selected proposal :

1. **Autonomous** trajectory generation for acquisition and tracking.
2. **Pointing constraint** consideration, target tracked only if feasible
3. Reference **frames removed**. Guidance works in Inertial Frame.
4. **Flexibility** to different target and operations.
5. **Agility** exploitation, actuation envelope considered in generation.
6. **Modularity** and compatibility with **single mode** AOCS.



Pros & cons

Guidance

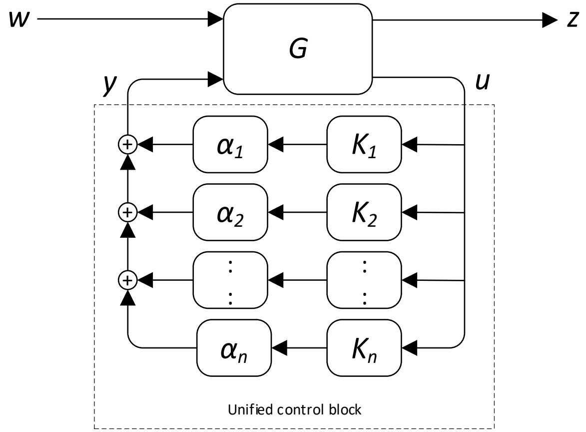
- ✓ Autonomy in trajectory generation
- ✓ Unique guidance frame: inertial
- ✓ Agility exploitation
- ✓ Explicit consideration of pointing constraints
- ✓ Prediction capabilities, relaxation of requirements on schedule
- ✓ Continuous smooth profiles, jumps avoided for control
- ✓ Autonomous generation of feasible feedforward control actions
- ✓ Adaptation to HW availability
- ✓ Mission extendibility, straightforward addition of targets
- ✓ Monitoring of unavoidable blinding events, minimization of exposure
- ✓ Target robustness, i.e., priority of constraints (tuneable)
- ✓ Reusability and flexibility to different constraints

- ❖ GPB needs to be designed for each mission
- ❖ Conservatism in actuation constraints for modularity
- ❖ Performance hindered by single mode implementation
- ❖ Computationally expensive
- ❖ More work required in blinding constraints

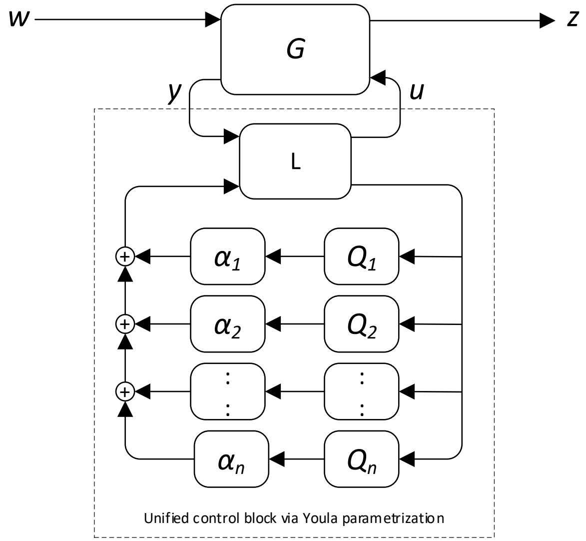
Control

Design approach (1/2)

- A single-mode AOCS requires a control solution which is capable of reconfiguring and/or adapting to different operating regimes without jeopardizing stability and while maximizing tracking and disturbance rejection performance as well as robustness to uncertainties.
- Several techniques were investigated such as linear parameter varying (LPV) control, classical adaptive control, and fault-tolerant and reconfigurable control.
- Systematic control design and analysis for these approaches is most adequately done within the LFT framework with a direct connection to the powerful robust control methodology.
- This framework offers the means to provide (global) guarantees for stability, robustness and performance which is highly relevant for a single-mode AOCS.
- Specifically, it was traded-off that the best way forward is:
  - To design controllers for specific tasks (e.g., slewing, pointing, safety)
  - Subsequently interconnect them via an intelligent switching scheme.
  - Can be done by means of the Youla-based control switching approach
  - Allows to re-parameterize controllers K_i as $L \star Q_i$, to obtain the control switching scheme $K = L \star Q$ with $Q = \alpha_1 Q_1 + \alpha_2 Q_2 + \cdots + \alpha_n Q_n$.
  - Here $\star$ denotes the star product between L and Q. The advantage of this approach is that the control switching scheme is guaranteed to be stable for any $\alpha_i \in [0,1]$, $\sum_{i=1}^{n} \alpha_i = 1$.



Unified control block



Unified control block via Youla parametrization

Control

Design approach (2/2)

Virtual sensors and actuators for-all-in-one mode AOCS

Design and reparameterization

3 controllers were synthesized: 1 for slewing, 1 for pointing (science), and 1 for safety. Reparametrizing the controllers into the Youla domain proceeds as follows:

- Let $G = ss(A,B,C,D)$ with (A,B) and (A,C) are stabilizable and detectable.
- Compute the coprime factors $G = NM^{-1} = \tilde{M}\tilde{N}^{-1}$, which satisfy the Bezout identity

$$\begin{pmatrix} \tilde{X} & -\tilde{Y} \\ -\tilde{N} & \tilde{M} \end{pmatrix}\begin{pmatrix} M & Y \\ N & X \end{pmatrix} = I$$

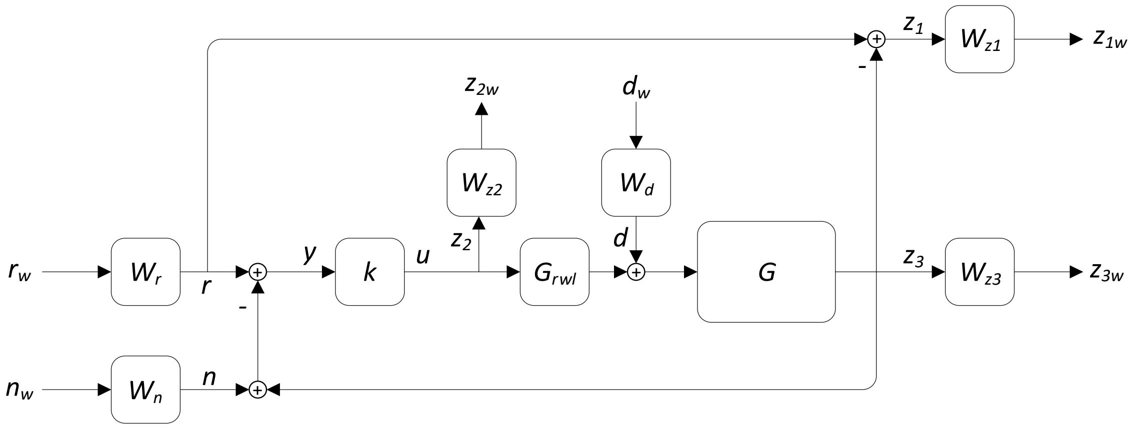
- Note: All 8 transfer matrices are in RH_∞. Can be done with IQClab.
- Now possible to obtain an entire family of controllers that internally stabilize G:

$$K = (Y - MQ)(X - NQ)^{-1} = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \star Q.$$

- Here Q can be any transfer matrix in RH_∞. This is the celebrated Youla parameter.
- For a give controller K that internally stabilized G, even if K is unstable, it is possible to obtain an L and Q through the coprime factors of $K = UV^{-1} = \tilde{V}^{-1}\tilde{U}$.
- Then the Youla parameter can be taken as

$$Q = -\left(\tilde{X}U - \tilde{Y}V\right)\left(-\tilde{N}U + \tilde{M}V\right)^{-1} = -\left(\tilde{X}K - \tilde{Y}\right)\left(-\tilde{N}K + M\right)^{-1}$$

- For the three designed controllers (K_1, K_2, K_3). These can then be reparametrized as described above to obtain the unified controller $L \star (\alpha_1 Q_1 + \alpha_2 Q_2 + \alpha_3 Q_3)$ with $\sum_{i=1}^{3} \alpha_i = 1$, $\alpha_i \geq 0$.



Implementation

- Implementation is straightforward, not different from the implementation of any other linear controller.
- Switching between controllers is taken care of by an internal function and based on size of control error and guidance angular rate. The safe controller is activated if the thresholds are exceeded. Every transition is done smoothly.
- The control block also includes a feedforward compensator as well as a ΔV controller.

Pros & Cons

Control

- ✓ Systematic and task specific design approach with guarantees for performance based on H_∞-synthesis and robustness analysis techniques
- ✓ Systematic integration of different task specific controllers based on Youla-parameterizations
- ✓ Smooth switching between configurations/tasks
- ✓ Guaranteed stability during switching
- ✓ Possibility to perform robustness analysis during switching
- ✓ Light on needed resources for implementation
- ✓ Can be integrated with fault tolerant control schemes
- ✓ Availability of powerful techniques and tools
- ✓ Compatible with powerful V&V approaches for certification

- ❖ Somewhat Higher order controllers (in terms of states)
- ❖ No flight heritage (although the implementation is just a state-space)
- ❖ Dedicated knowledge required

Configuration manager

Design summary

The all-in-one AOCS requires a manager capable of ensuring an autonomous and continuous performance of the system by means of the following tasks:

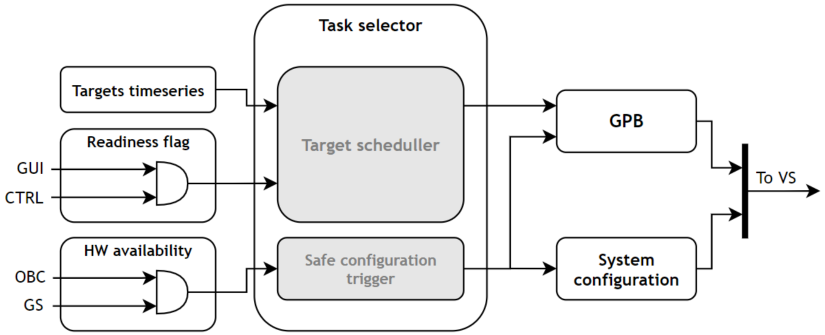
- **Information manager**: Receive information from VS, GUI, CTRL and VA, and manage it to determine system status
- **Status checker:** Continuously check sensors and actuators availability
- **Target scheduler:** Autonomously command target considering received timelines and current availability status
- **Pointing performances:** Provide required data for performing autonomous attitude control and pointing
- **Safe configuration triggering:** Autonomously determine if the spacecraft should go into safe configuration.

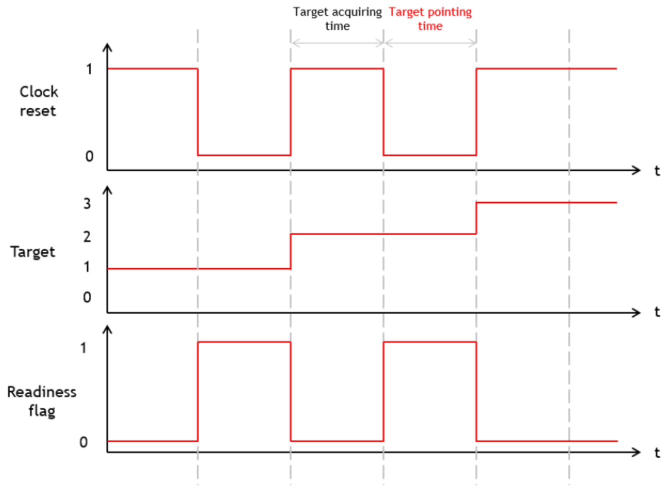
The achievement of these functions is due to a flags-based structure:

- ○ Readiness flag (GUI+CTRL): triggers target pointing considering GUI status and current CTR
- ○ HW availability flags (SEN & ACT): determine operative status of units by comparing OBC and ground station status sources
- ○ System configuration flags (VS & GUI): command system configuration

Task selector function schedules and sets the operation configuration given a timeline of desired targets and performances. Two types of timelines are allowed:

$$\text{Timeline} = \begin{bmatrix} \text{Target1} & \text{Target 2} & \dots & \text{Target } N \\ \text{Pointing time} & \text{Pointing time} & \dots & \text{Pointing time} \\ \Delta V_1 & \Delta V_2 & \dots & \Delta V_N \end{bmatrix} \quad \text{or} \quad \text{Timeline} = \begin{bmatrix} \text{Target1} & \text{Target 2} & \dots & \text{Target } N \\ \text{Starting time} & \text{Starting time} & \dots & \text{Starting time} \\ \Delta V_1 & \Delta V_2 & \dots & \Delta V_N \end{bmatrix}$$





Pros & Cons

Configuration manager

Operational:

- ✓ Replaces modes by tasks, providing greater flexibility and possibility of mission upgrade with new compatible objectives.
- ✓ Automatic execution of target schedules.
- ✓ FDIR functionalities encapsulation.

V&V:

- ✓ Validated at unit level
- ✓ Facilitates testing and V&V of the rest of the system.

Reusability:

- ✓ Can be extended as required.
- ✓ Can be reused as a framework.

Operational:

- ❖ Complexity of FDIR operations limited to compatibility with the all-in-one AOCS architecture, i.e., limited to target request and reconfigurations.

V&V:

- ❖ Response needs to verified at system level.

Reusability:

- ❖ Interfaces with the rest of the GNC shall be maintained.
- ❖ Missionization requires more work than just parametrization.

Test results

Summary of test campaign

- **Simulator development**
- **Test campaign (8 tests):**
    - **PERF01:** Maneuverability and pointing performance during limb-looking activities (100 sims, parametric perturbations)
    - **PERF02:** Maneuverability and pointing performance during occultation observations (100 sims, parametric perturbations)
    - **PERF03:** Propulsive manoeuvring performance (200 sims, parametric perturbations)
    - **LEOP01:** Detumbling after launcher separation (50 sims, parametric perturbations)
    - **FDIR01:** Robustness to sensor outages/failures (1 sim)
    - **FDIR02:** Robustness to actuator outages/failures during science (1 sim)
    - **FDIR03:** Robustness to actuator outages/failures during propulsive manoeuvres (1 sim)
    - **FUNC01:** Functional test for verifying secondary guidance profiles and ancillary services (100 sims, parametric perturbations)
- **Next:** Selected overview of results and comparison to ALTIUS mission

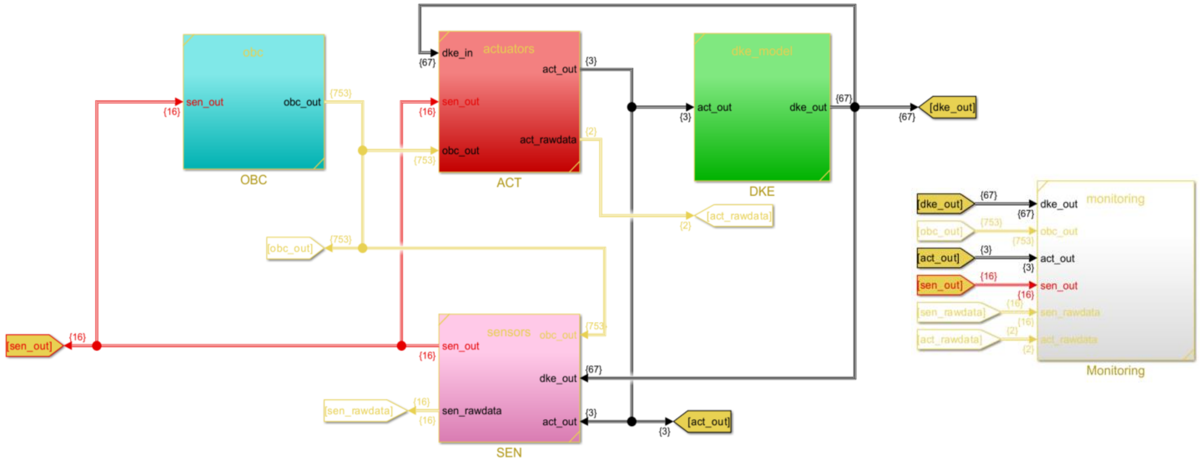
Simulation environment
Design summary

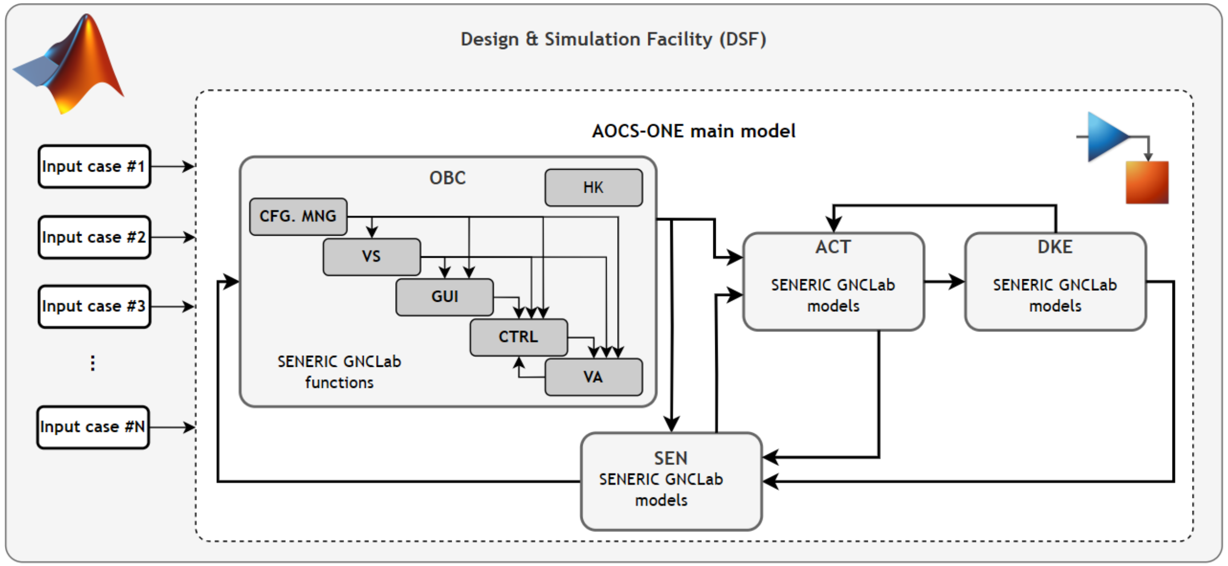
AOCS-ONE simulation environment is based on DSF (Design & Simulation Facility), an in-house tool for supporting the design and development of simulation environments. The main functions it provides are:

- An intuitive model-based framework built upon validated models
- Implementation and customization of different use cases or scenarios
- Flexibility to customize input and output profiles for each model and scenario for tailored and precise simulation results
- Running scenarios in normal/accelerator/rapid-accelerator mode
- Executing batches of simulations, either individually or in parallel, to optimize efficiency and streamline outputs
- Running MC simulations by adjusting specific input parameters with targeted distributions for precise and optimized outcomes
- Flexibility to implement customized post-processing functions for outputs analysis

To streamline the development of the simulation environment, validated models and functions were used, ensuring efficiency and reliability in the design and testing phase.

AOCS-ONE Simulator




Design & Simulation Facility (DSF)

AOCS-ONE main model

Input case #1
Input case #2
Input case #3
⋮
Input case #N

OBC
HK
CFG. MNG
VS
GUI
CTRL
VA
SENERIC GNCLab functions

ACT
SENERIC GNCLab models

DKE
SENERIC GNCLab models

SEN
SENERIC GNCLab models

Test results and comparison with baseline

Some notes on functional performance

It was argued that a single-mode AOCS requires a 3-axis stabilization strategy (mostly to ensure a genuine single-mode AOCS architecture)

- We developed a different safe mode that relies on 3-axis stabilization rather than the Bdot modes that are considered in the ALTIUS AOCS.
- Likely possible to include spin-stabilized solutions too (potentially at the price of having a less genuine single-mode AOCS architecture).

Also observed limitations of the single-mode AOCS during the rate-damping after launcher separation

- The single mode AOCS always tracks attitude errors, and not rate errors. Makes rate damping with initial rates of 5 deg/s difficult.
- Also, here likely possible that the solution can be generalized to include a proper rate damping functionality.

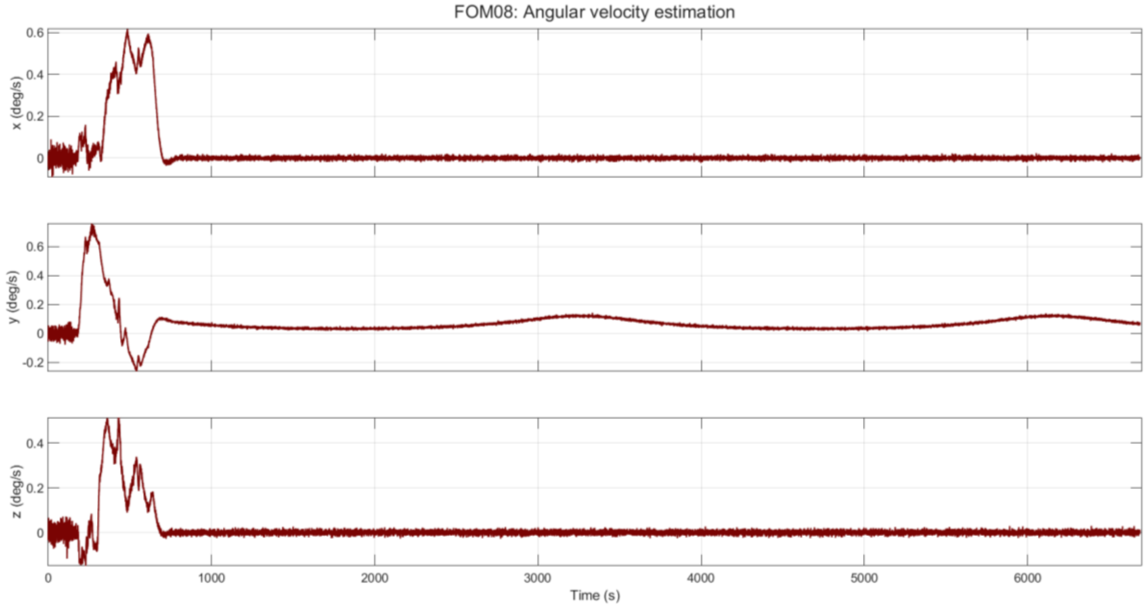
The single-mode AOCS is highly flexible in terms of configuration management, and reconfiguring to any HW availability. In the baseline AOCS, this would potentially require including additional modes to be able to cover the same functionality

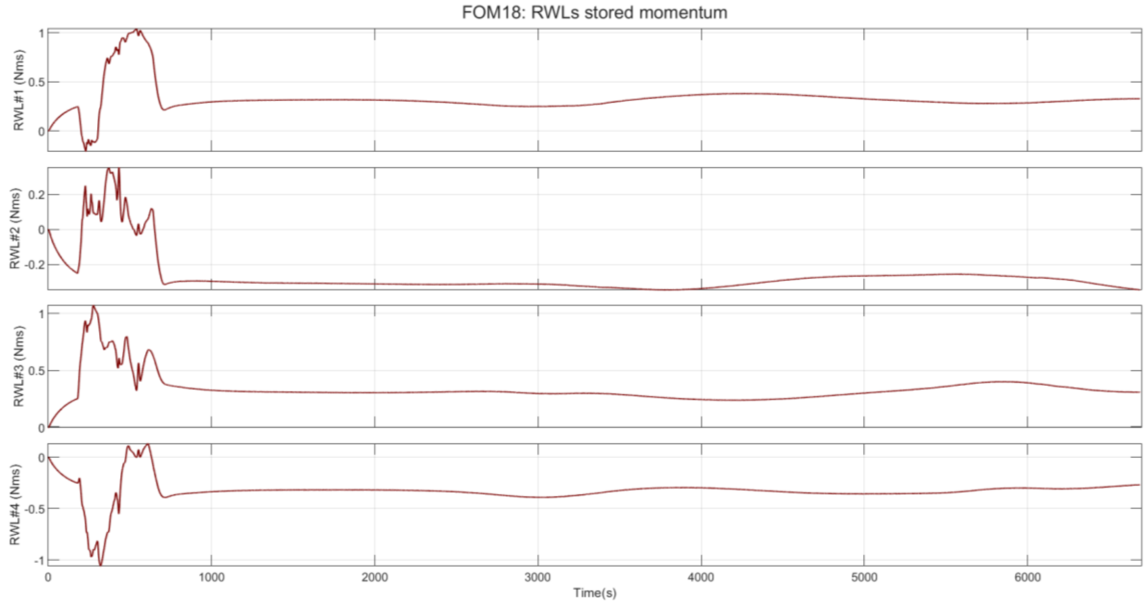
- This makes the single-mode AOSC very robust to HW failures. It was demonstrated that the single-mode AOCS could continue to operate in severely degraded scenarios. This is further addressed later.
- This increases the autonomy of the system.

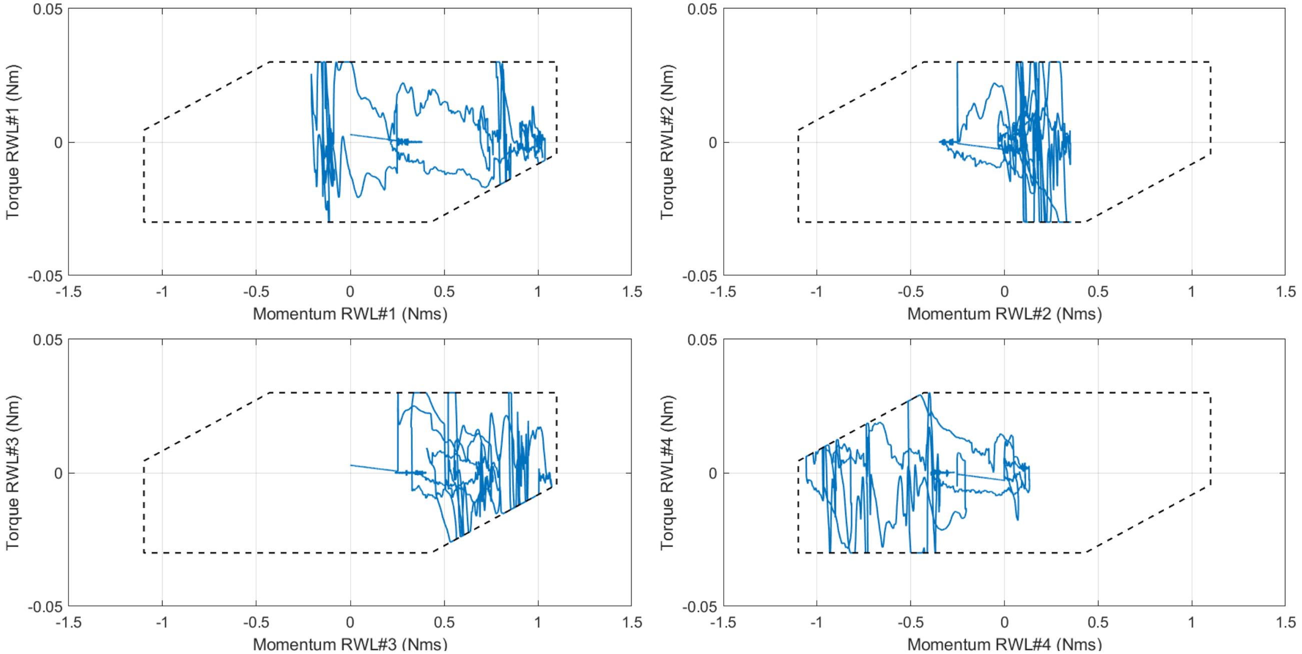
Test results

Detumbling performance (LEOP01)

- Due to the 3-axis stabilization constraint, the initial tumbling velocities were reduced from $5\ \mathrm{deg}/s$ to $1\ \mathrm{deg}/s$
- The AOCS-ONE successfully detumbles the SC without using THRs, although the RWL torque commands start to saturate they reach normal levels once stabilized and the VA manages to desaturate the wheels once the velocity is reduced.





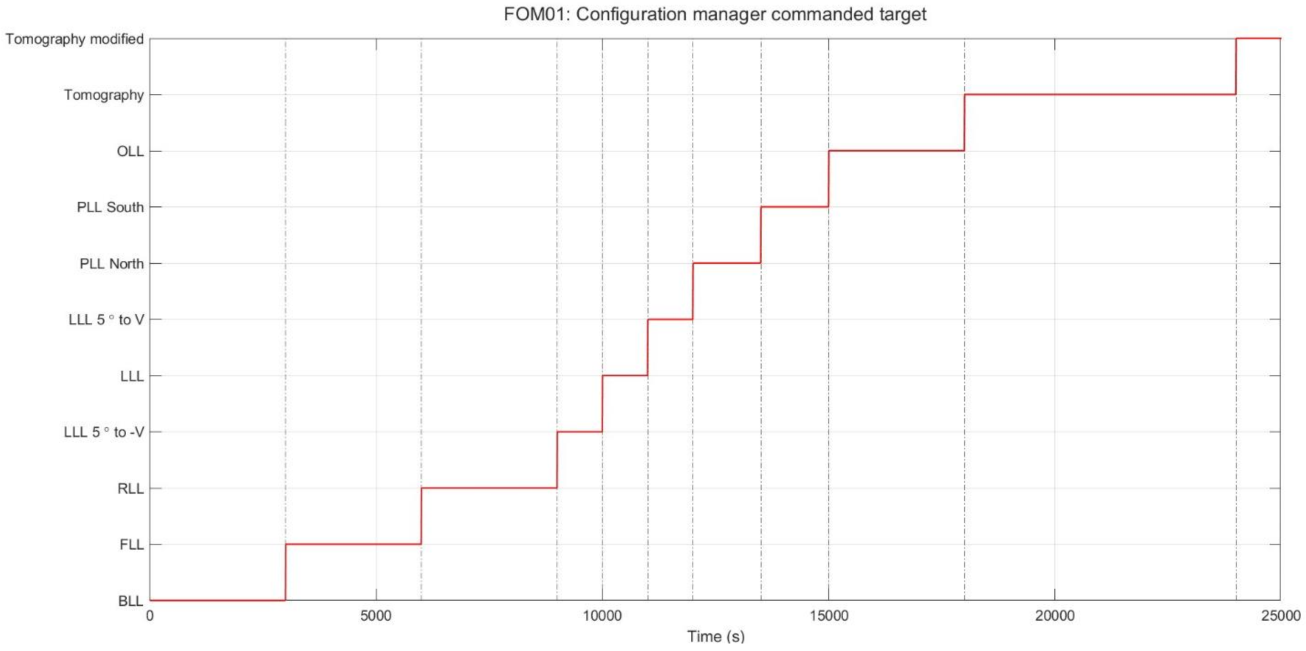


Test results and comparison with baseline

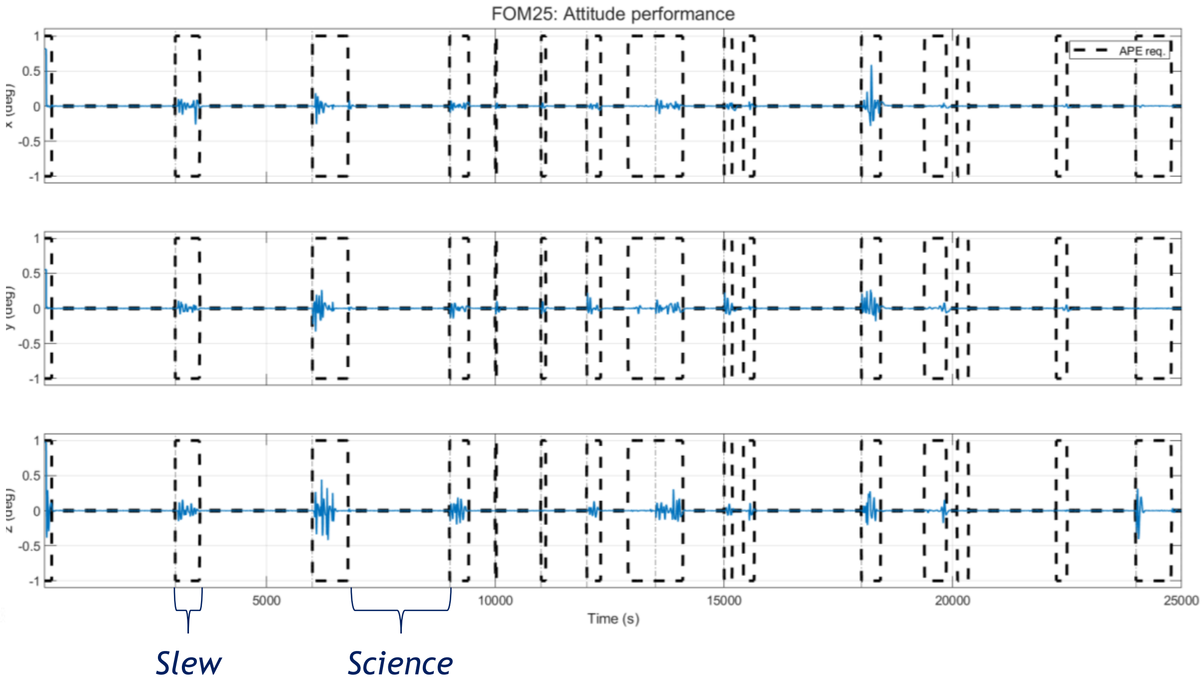
Agility (slewing) performance (1/2) (PERF01)

- Achieving agility requirements was equally challenging for both ALTIUS baseline and AOCS-ONE designs
- Limiting factor is the RWL torque-momentum envelope which in turn limits the control bandwidth
- For celestial targeting (PERF02) not all targets could be reached (also not for baseline). By further tuning the results may be (slightly) improved
- The GUI and VA ensure fast tracking and RWL torque-momentum envelope is not exceeded, and the CTR can track the GUI

PERF01: Configuration manager commanded target



PERF01: Nominal simulation attitude APE

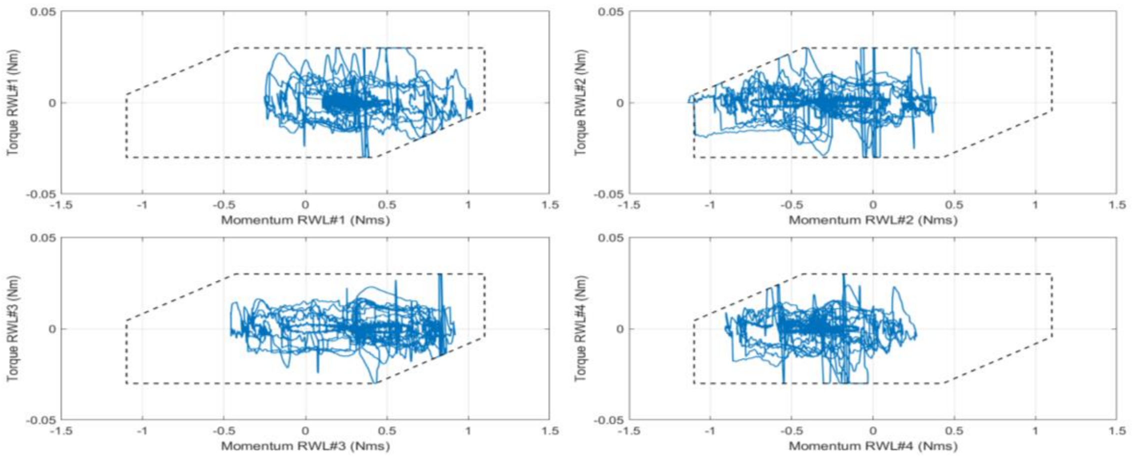


Test results and comparison with baseline

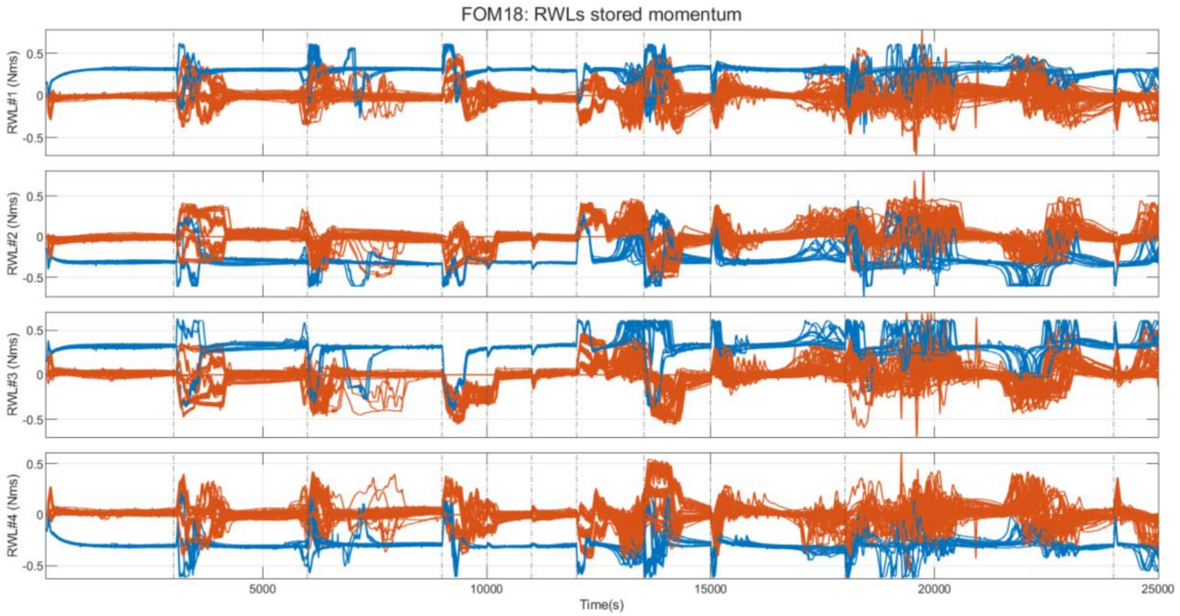
Agility (slewing) performance (2/2) (PERF02)

- Achieving agility requirements was equally challenging for both ALTIUS baseline and AOCS-ONE designs
- Limiting factor is the RWL torque-momentum envelope which in turn limits the control bandwidth
- For celestial targeting (PERF02) not all targets could be reached (also not for baseline). By further tuning the results may be (slightly) improved
- The GUI and VA ensure fast tracking and RWL torque-momentum envelope is not exceeded, and the CTR can track the GUI

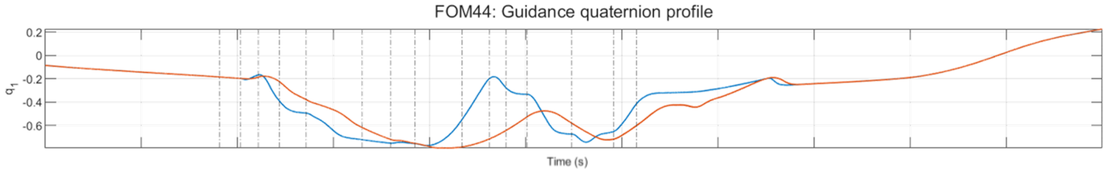
VA RWL envelope exploitation in nominal simulation



VA continuous desaturation with 4 (blue) and 3 (orange) RWLs available



Guidance trajectory with 4 (blue) and 3 (orange) RWLs available

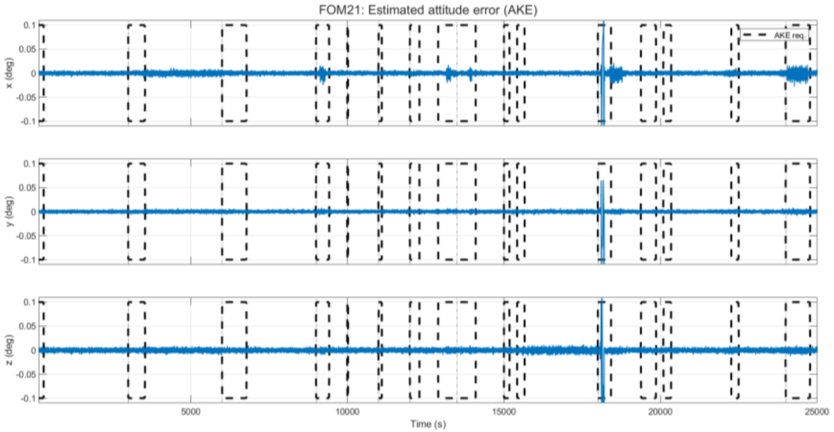


Test results and comparison with baseline

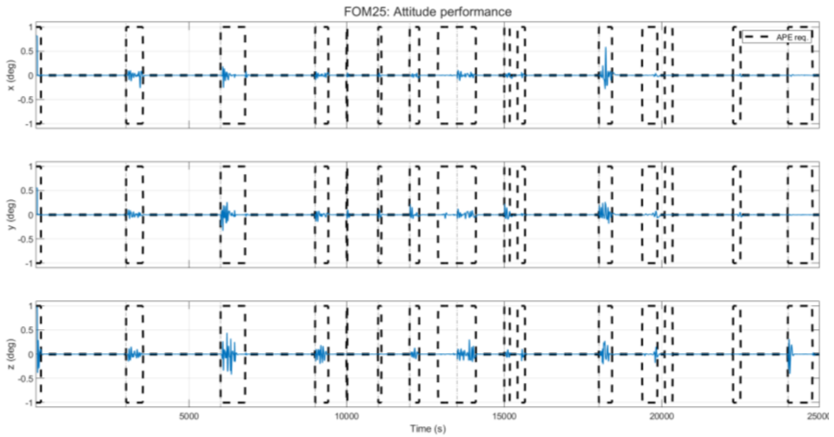
Pointing performance (PERF01)

- The (AKE, APE, RPE) pointing performances are also similar.
- The difference was in the considered STR noise levels, baseline [0.9,1.6,13.5] arcsec and AOCS-ONE [3,3,30] arcsec.
- Did not perform an exhaustive tuning of the single-mode AOCS -> Primary focus was functional performance. A dedicated fine tuning with the correct noise levels would yield similar results.

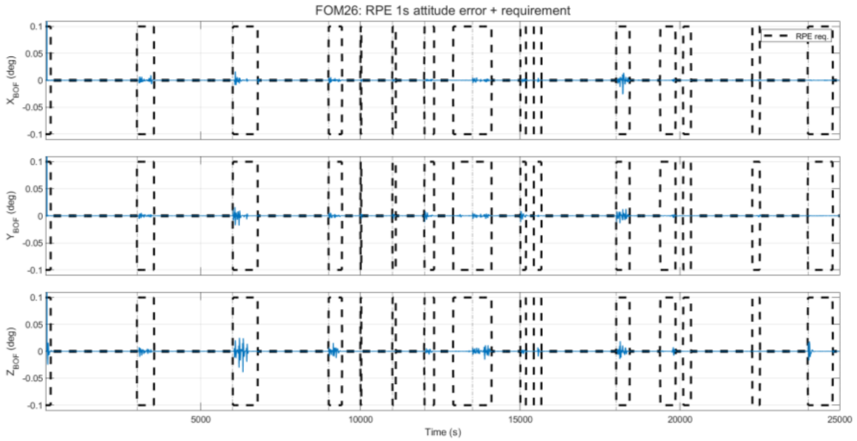
PERF01: Nominal simulation attitude AKE



PERF01: Nominal simulation attitude APE



PERF01: Nominal simulation attitude RPE 1s



Test results and comparison with baseline

Fuel consumption

There is no relevant difference in terms of fuel consumption:

- This is because of the thruster layout (i.e., there are only 4 thrusters located in the x-y plane)
- However, the VA minimizes the fuel consumption by design.
- Hence, even if there is no noticeable reduction in fuel consumption with respect to the benchmark mission, there would be a potential improvement for missions with more complex thruster configurations.
- Moreover, the VA is a generic solution that can be easily adapted to any mission, which would reduce the engineering effort.

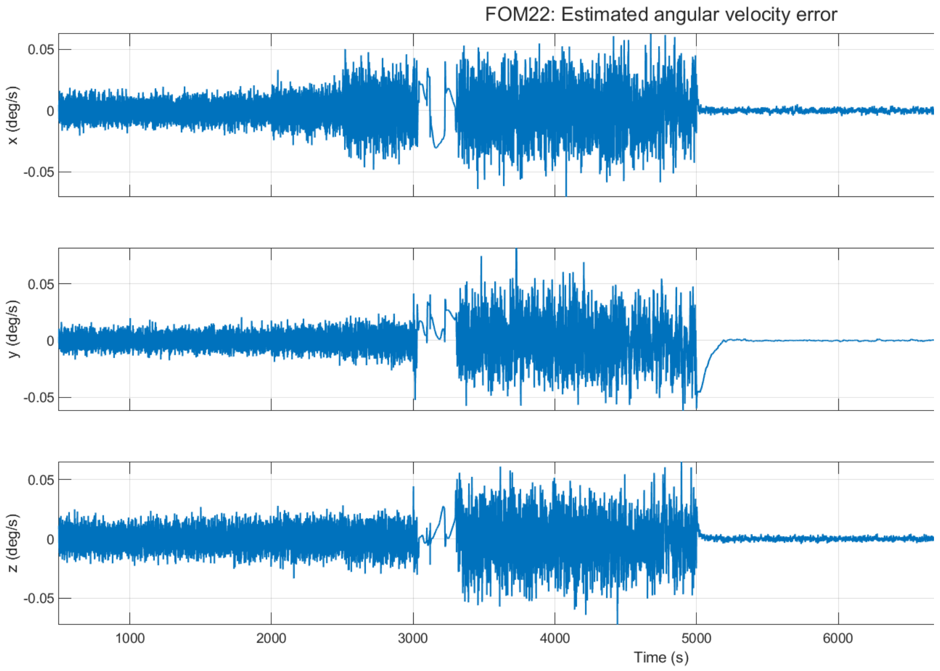
Test results and comparison with baseline

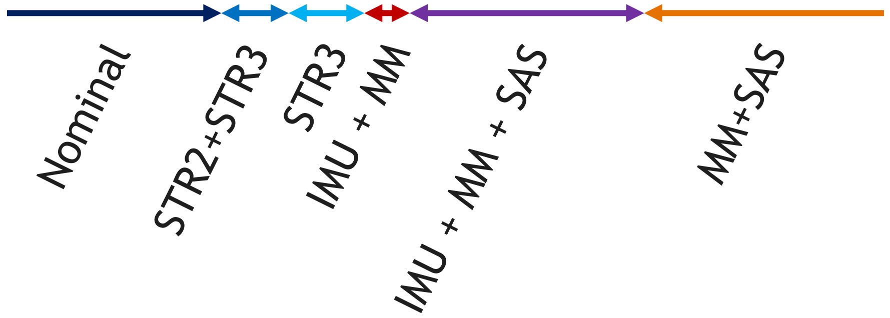
FDIR01 - sensor outages (1/4)

Description	ALTIUS baseline response	AOCS-ONE response
Test sensor outages 1. Reach a Limb Looking attitude and with full HW availability. 2. At $t_1 - t_7$ resp. inject (cumulative) outage for: STR1, STR2, STR3, GNSS-1, GNSS-2, IMU-1, IMU-2.	• The AOCS reaches Limb Looking mode • Once the 3rd STR fails, the attitude is propagated for about 1min, then a safe mode reconfiguration is triggered (augmented Bdot algorithm). • As GNSS and IMU are not used in safe mode, the outage does not have any effect. • Assuming the S/C is put back in Limb Looking mode after the safe mode reconfiguration, the GNSS outage would result in on-board orbit determination to switch to TLE (as long as TLE age is not exceeding ~2weeks). • The IMU outage doesn't have a noticeable effect in Limb Looking mode (if STR are available).	• The AOCS reaches Limb Looking mode • Once the 3rd STR fails, the safe configuration is triggered which means that a slew is performed towards the Sun-bathing frame (SBF). • Subsequently, also the GNSSs and the IMUs become unavailable. • The VS then continues operating by means of the SAS, MMs and TLEs for attitude and position/velocity determination in combination with the propagation model that is embedded in the VS

Test results and comparison with baseline

FDIR01 – sensor outages (2/4)





Nominal

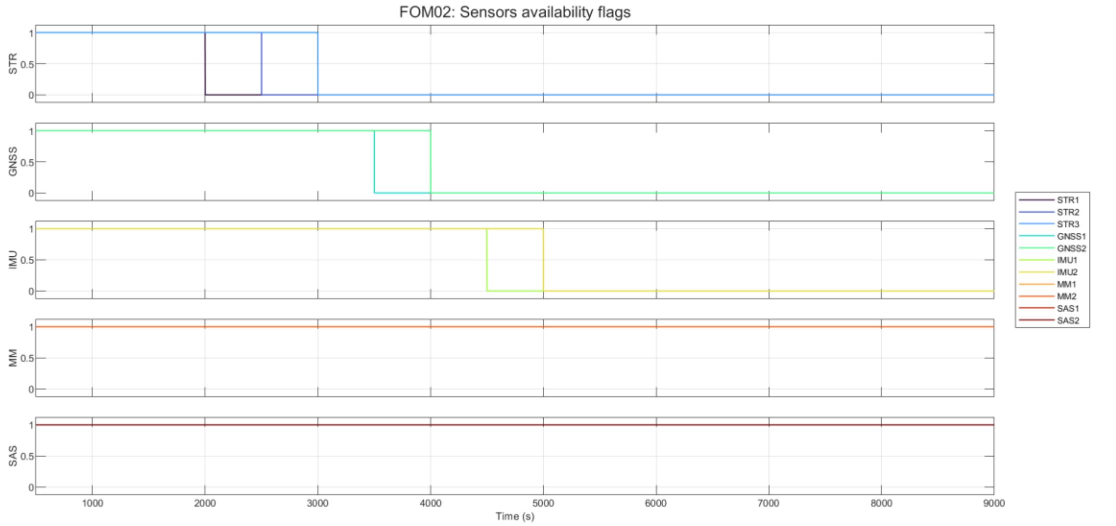
STR2+STR3

STR3

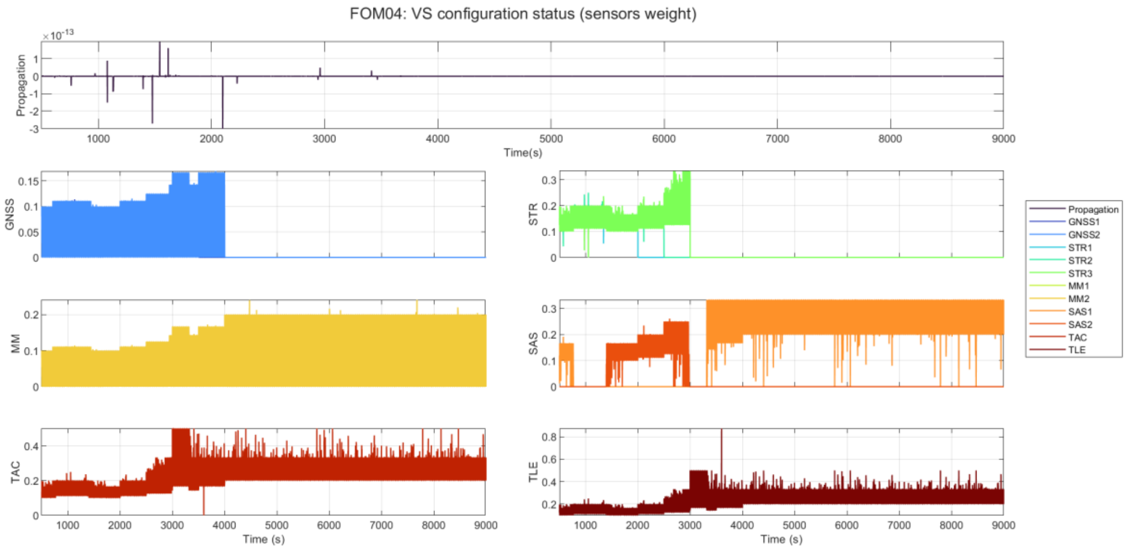
IMU + MM

IMU + MM + SAS

MM+SAS

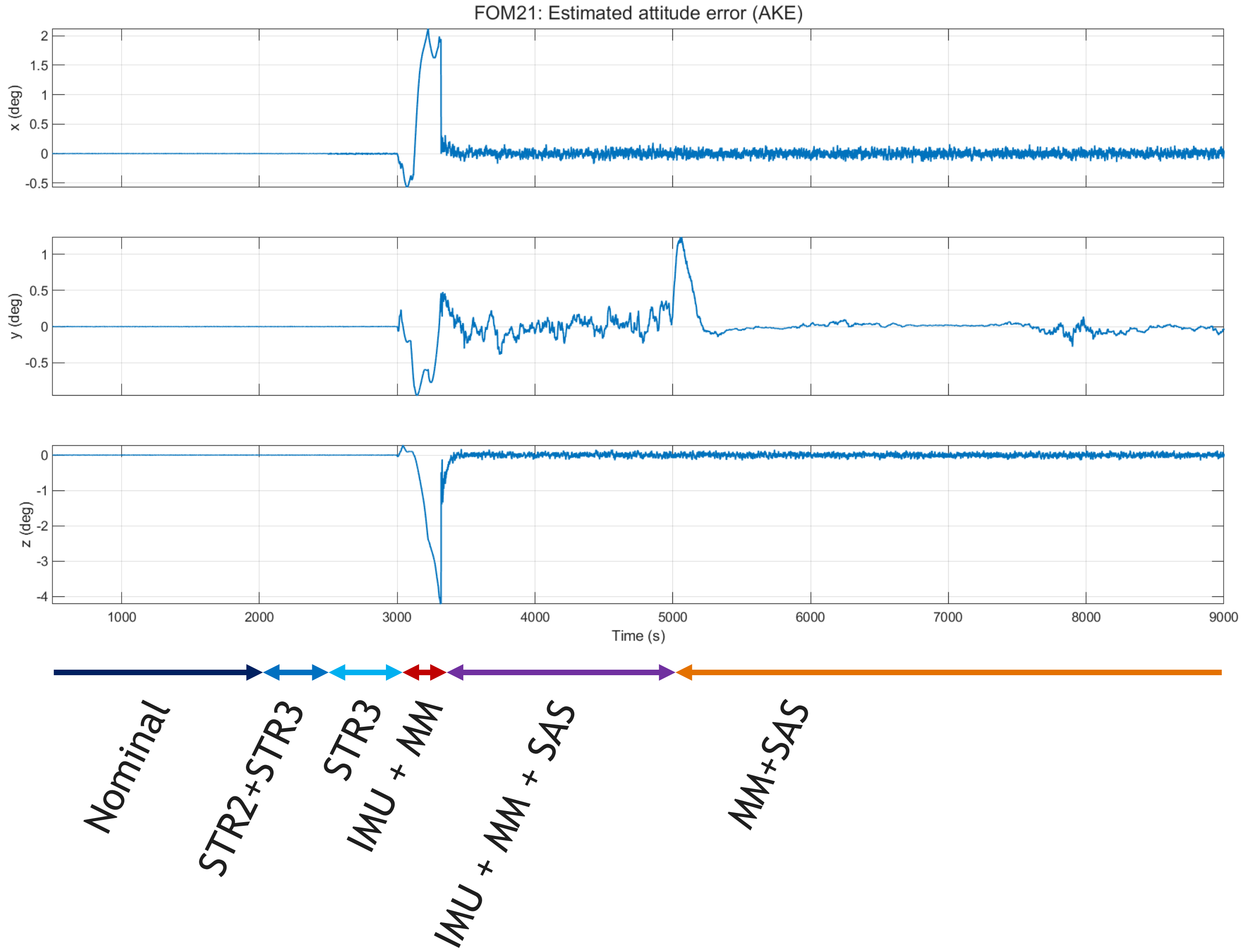


Autonomous reconfiguration through sensor weighting



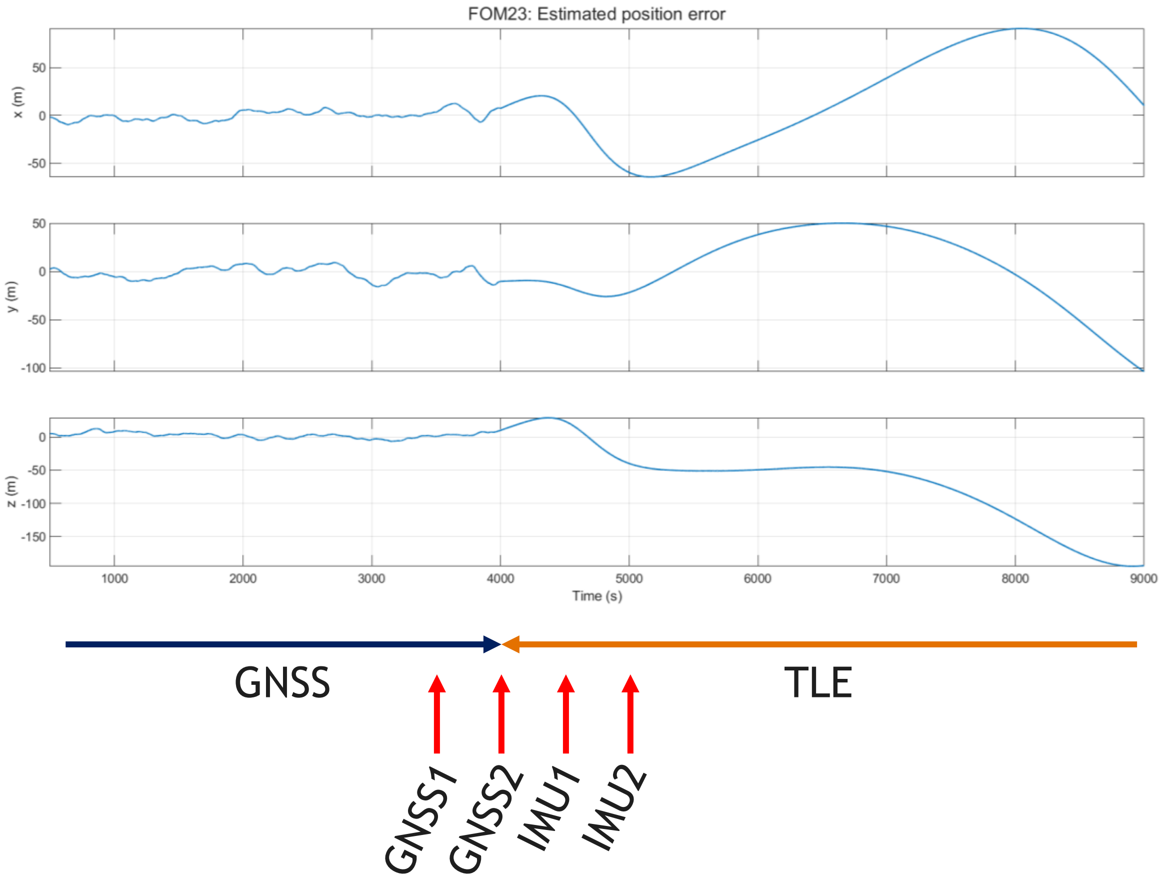
Test results and comparison with baseline

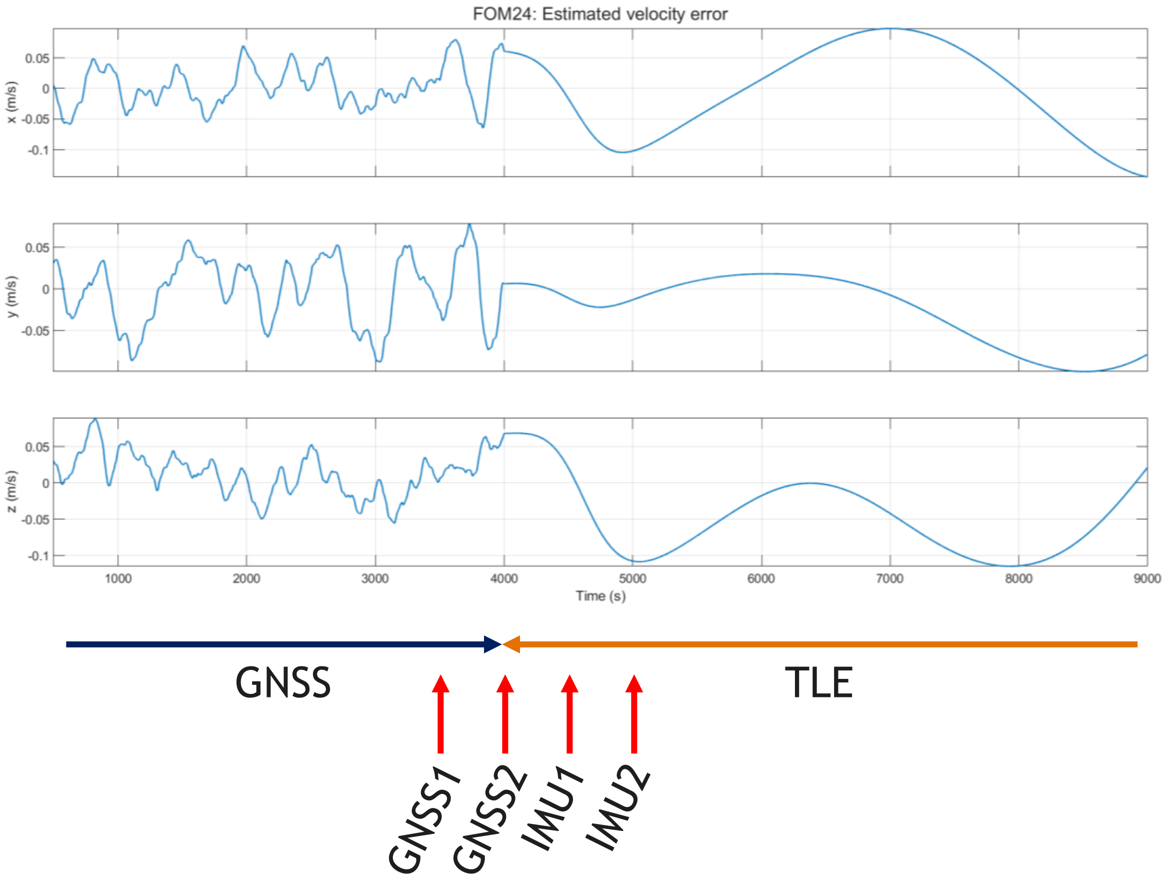
FDIR01 – sensor outages (3/4)



Test results and comparison with baseline

FDIR01 – sensor outages (4/4)





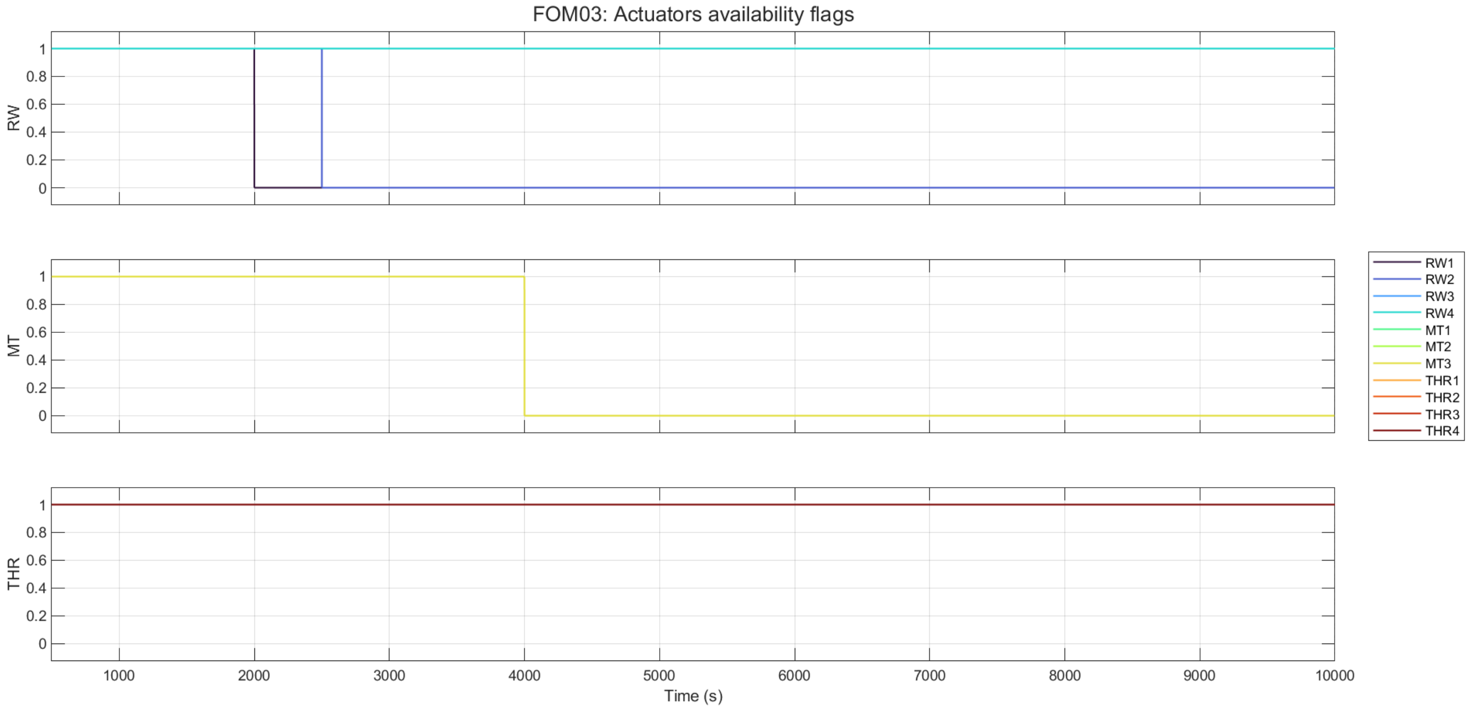
Test results and comparison with baseline

FDIR02 - actuator outages during pointing (1/3)

Description	ALTIUS baseline response	AOCS-ONE response
Test actuator outages during pointing 1. Reach a Limb Looking attitude and with full HW availability. 2. At $t_1 - t_3$ resp. inject (cumulative) outage for: RWL-1, RWL-2, all MTs. Note: Thrusters can be used for attitude control.	• The AOCS reaches Limb Looking mode. • Once the 1st RWL fails, it continues operating with the remaining 3 RWLs. • After the second RWL failure, safe mode reconfiguration is triggered by system FDIR. • Two options exist: o either the remaining two RWLs are a valid set to support safe mode, or they are not. o In the latter case, safe mode reconfiguration will fail and an OBC reboot will occur. Upon repeated failure to enter safe mode, an OBC lane switch-over will be commanded. Upon persistent double RWL failure, mission will be lost unless ground finds a solution. • Assuming safe mode is recovered with 2 remaining RWLs, the MT failure will trigger a new safe mode reconfiguration with the redundant RTU/MT coils.	• The AOCS reaches Limb Looking mode. • Once the 1st RWL fails, it continues operating with the remaining 3 RLWs. • After the 2nd RWL fails, the AOCS continues using the 2 remaining RWLs in combination with the THRs to realize 3-axis stabilization.

Test results and comparison with baseline

FDIR02 - actuator outages during pointing (2/3)



After RWL#1 and RWL#2 failed, the system switches to Sun-bathing frame, which is successfully acquired by using the remaining 2 RWLs and the THRs.

Test results

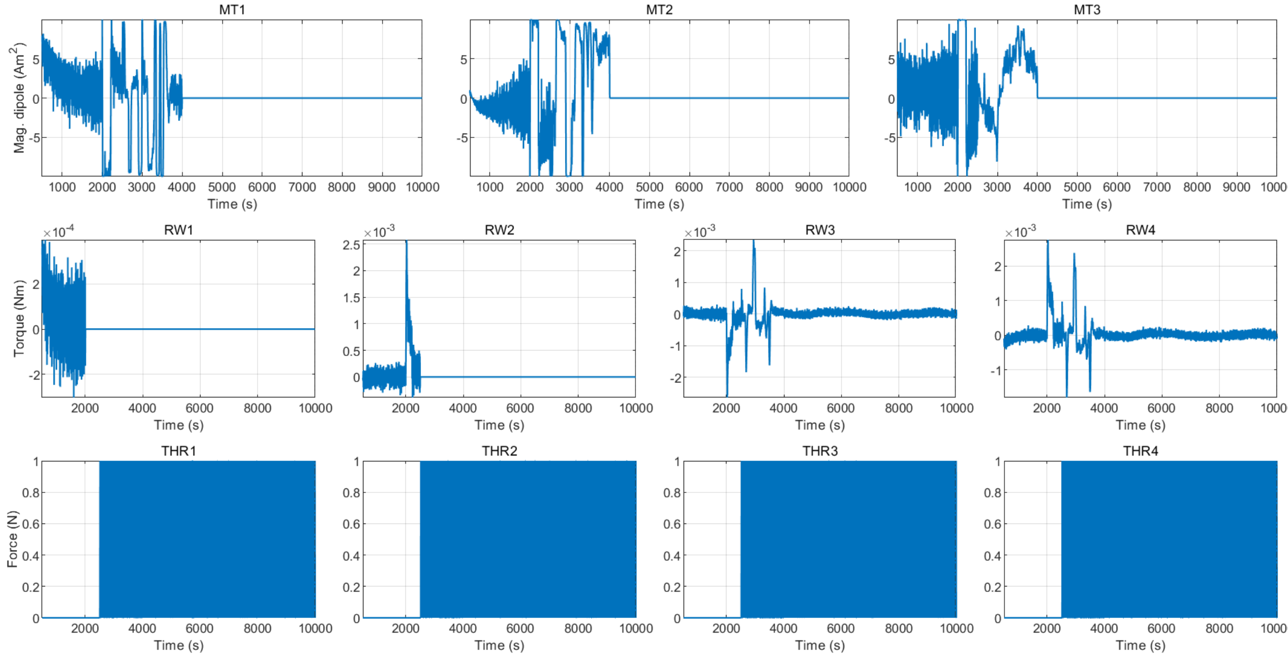
FDIR02 - actuator outages during pointing (3/3)

- After RWL#1 and RWL#2 failed, the system switches to Sun-bathing frame, which is successfully acquired by using the remaining 2 RWLs and the THRs.
- The VA reconfigures and start using thrusters.
- The APE remains (relatively small)

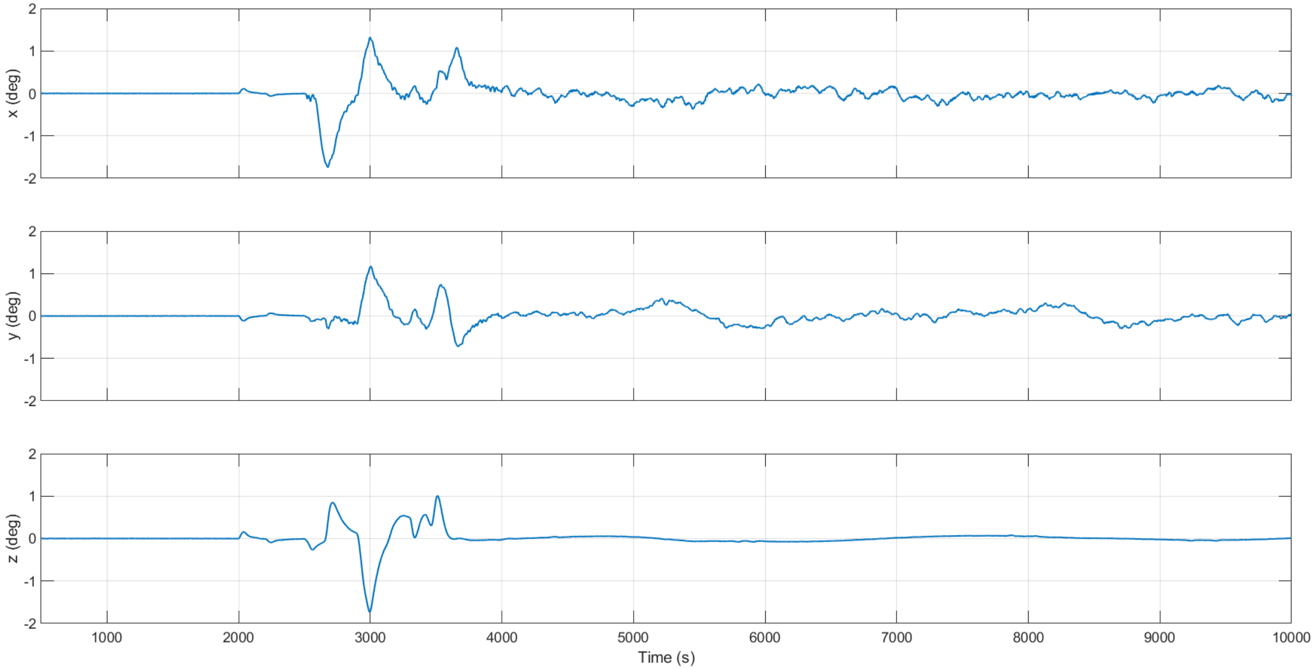
Note: Can only be sustained for a short period due to limited fuel supply

Note: Using thrusters implies propulsive action in this benchmark.

FOM05: VA configuration status (commanded actuation)



FOM25: Attitude performance



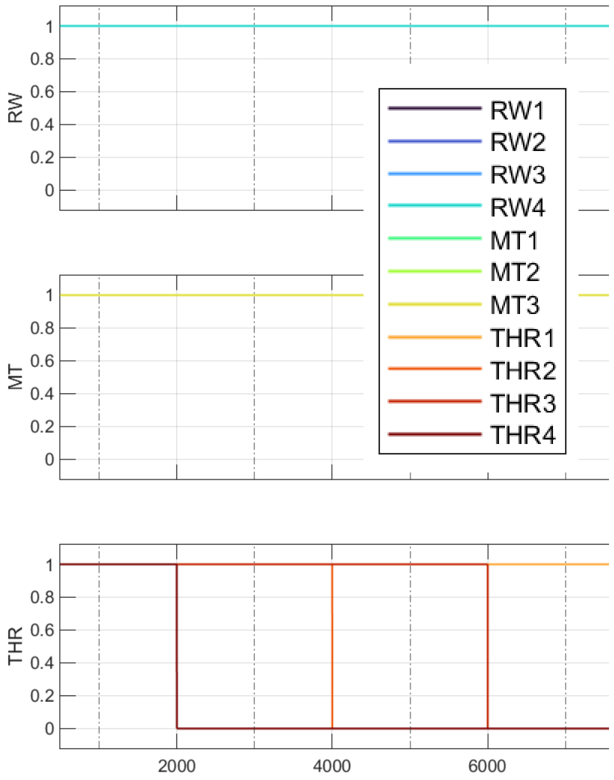
Test results and comparison with baseline

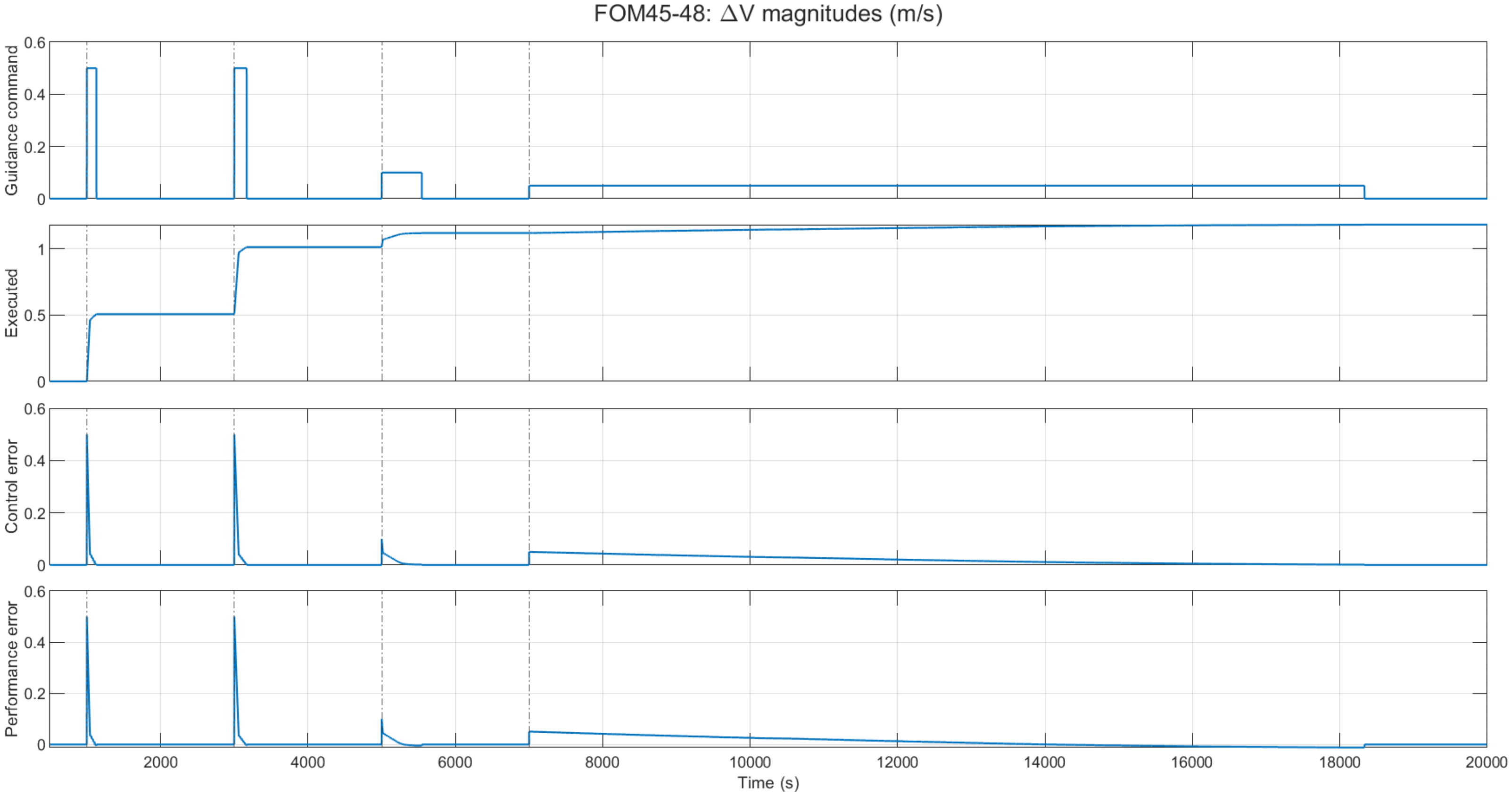
FDIR03 – actuator outages during propulsive manoeuvres (1/2)

Description	ALTIUS baseline response	AOCS-ONE response
Test actuator outages during ΔV 1. Reach ΔV attitude with full HW availability and perform a $0.5\ m/s\ \Delta V$ manoeuvre 2. At t_1 inject an outage on THR-1 and perform a $0.5\ m/s\ \Delta V$ manoeuvre 3. At t_2 inject an outage on THR-2 and perform a $0.05\ m/s\ \Delta V$ manoeuvre 4. At t_3 inject an outage on THR-3 and perform a $0.05\ m/s\ \Delta V$ manoeuvre 5. At t_4 inject an outage on all 3 MTs and perform a $0.05\ m/s\ \Delta V$ manoeuvre	The AOCS first reaches the FLT frame and respectively performs the following orbital raises: • $0.5 m/s$ with 4 THRs • $0.1 - 0.5 m/s$ with 2 THRs (2 valid THRs on a diagonal are used; achievable delta-V before wheel saturation depends on thruster alignment and CoM location) • $0.05 m/s$ with 2 THRs or no propulsion (depending whether remaining 2 THR are on same diagonal) • FLT mode is maintained, but propulsion manoeuvre is not performed • FLT mode reconfiguration is triggered upon MT failure, leading to switch to redundant RTU/MT coils, no propulsive manoeuvre is executed Note that it would be ground responsibility to trigger a switch-over to the redundant RTU to exercise the redundant propulsion drivers. This is not done autonomously on-board. Short 0.05m/s manoeuvres can in principle be achieved with a single thruster, but this would be done via dedicated ground commands.	The AOCS first reaches the FLT frame and respectively performs the following orbital raises: • $0.5 m/s$ with 4 THRs • $0.5 m/s$ with 3 THRs • $0.05 m/s$ with 2 THRs • $0.05 m/s$ with 1 THRs • $0.05 m/s$ with 1 THRs & no MTs. The VA ensures that the generated thrust is smeared out over a larger time window in case the RWLs reach their saturation level. This differs from ALTIUS baseline, where manoeuvre is stopped if APE exceed a certain value due to cluster saturation to prevent further error increase. The VA, instead, keep operating without inferring in an APE increase, autonomously deciding thrust times and desaturation times based on cluster status. Alternatively, the RWLs can be desaturated first and then perform a ΔV manoeuvre with sufficiently limited magnitude to prevent RWL saturation.

Test results and comparison with baseline

FDIR03 – actuator outages during propulsive manoeuvres (2/2)





Propulsive manoeuvres are performed with 4, 3, 2, and 1 THRs respectively. For the last manoeuvre with 1 THR the RWLs are already nearly saturated. Then the VA smears out the execution over a longer period. Alternatively, the RWL can (should) be off-loaded first.

Test results

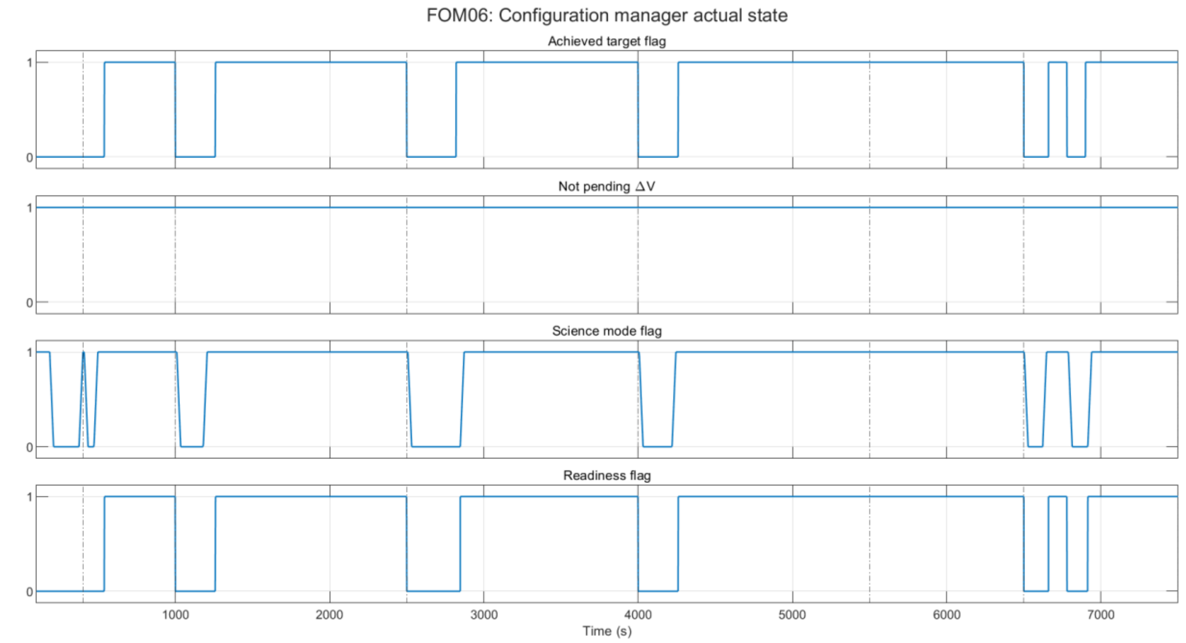
Ancillary services: acquiring targets

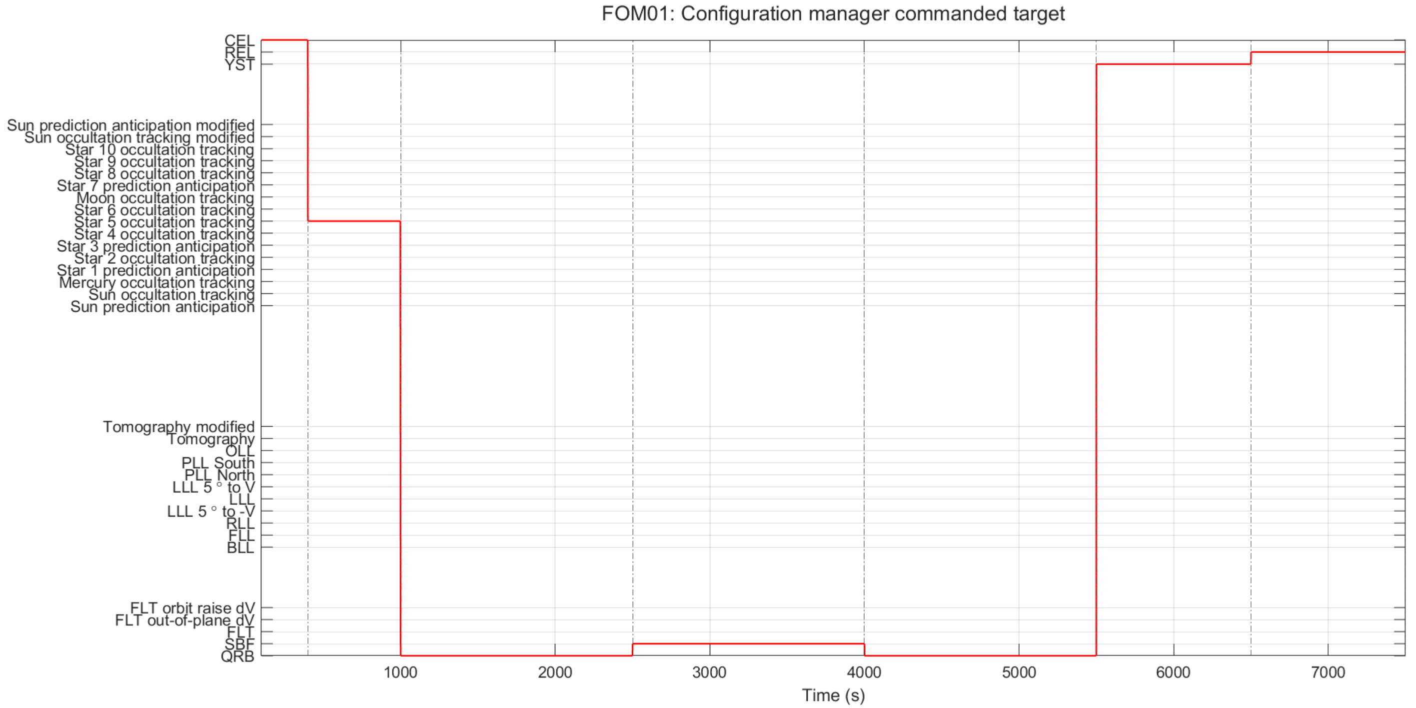
Target sequence:

1. CEL: inertial target (Star 5)
2. Star 5 occultation tracking (constant velocity around one axis)
3. QRB
4. SBF (sun-bathing frame)
5. QRB = GED
6. YST
7. REL: ground station target

All targets are achieved in time, with smooth slews and small stabilization times (science triggering close to guidance target achieved flag rise).

The small slew produced at the last target might be produced by a pointing constraint avoidance.





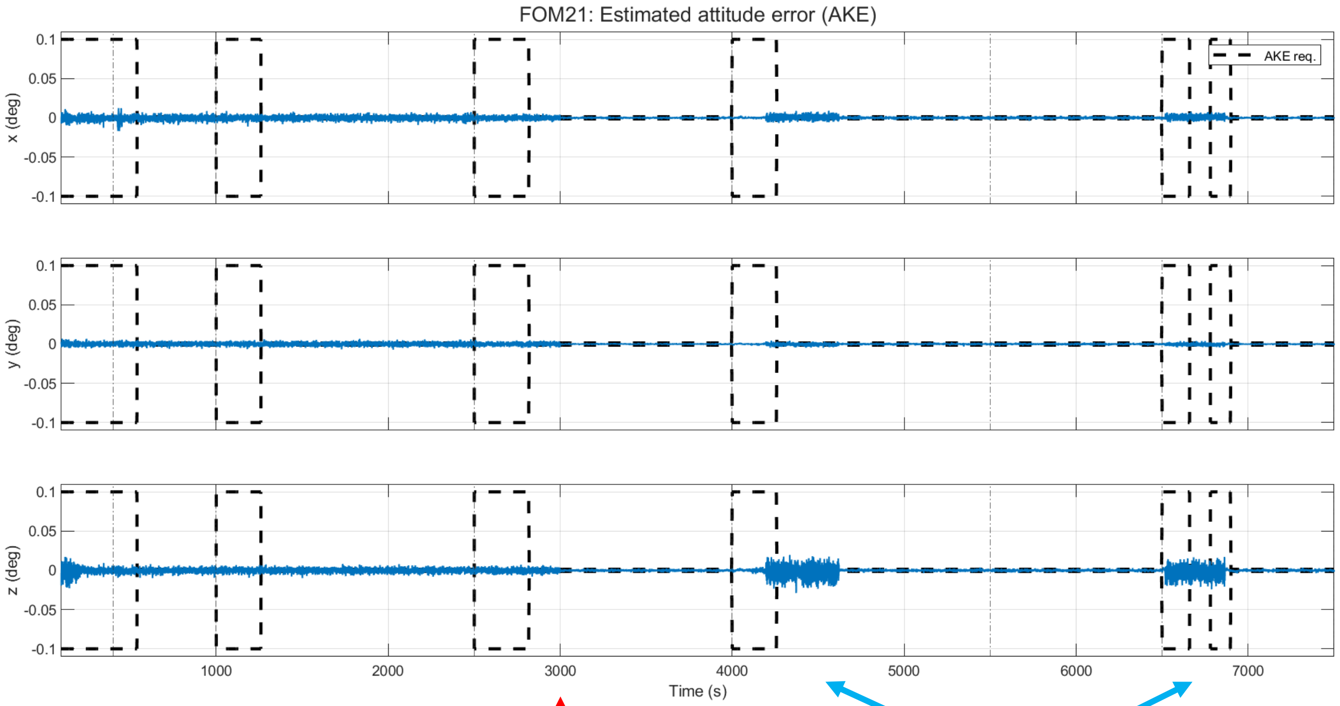
Test results

Ancillary services: pointing performance

The AKE is within requirements while IMUs are available, whereas the error during slews after the IMU outages is larger.

This is due to the angular velocity estimation, for which we see two performance regions:

- With IMUs: noisy but unbiased.
- Without IMUs, driven by dynamic model propagation and attitude sensors (mainly STRs): little to no noise, but degraded performance during slews (see seconds 6500-7000s).



IMU outages

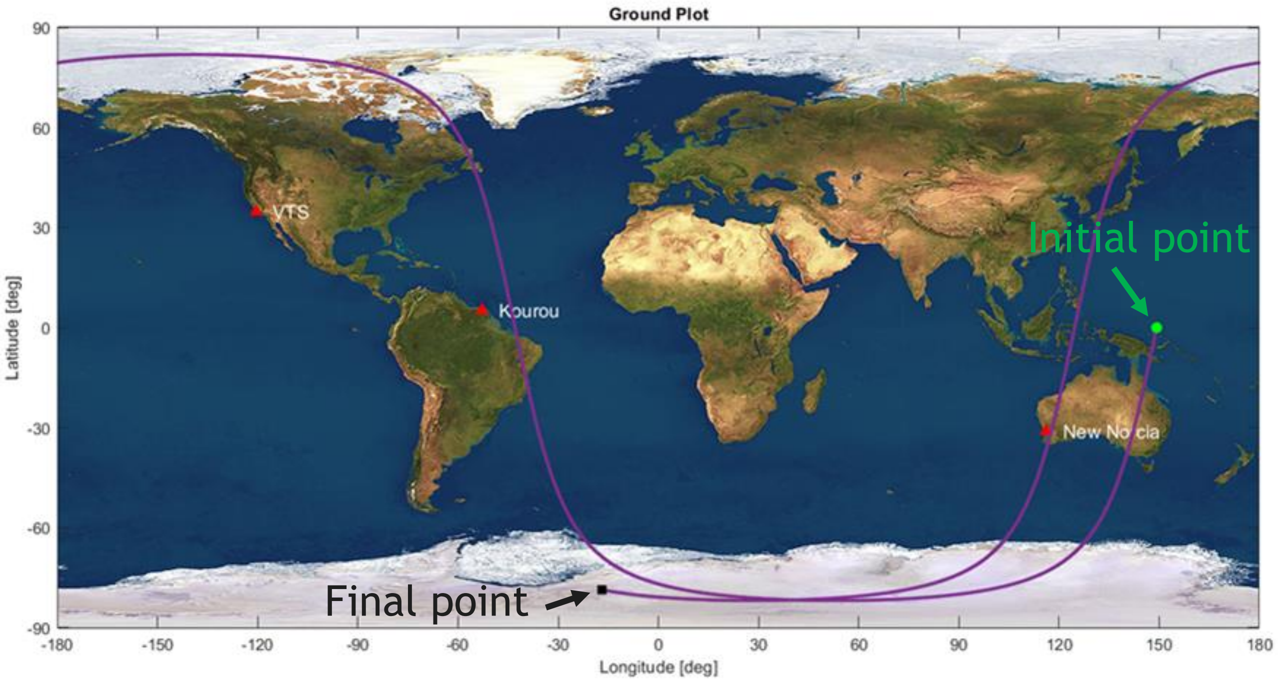
Slews

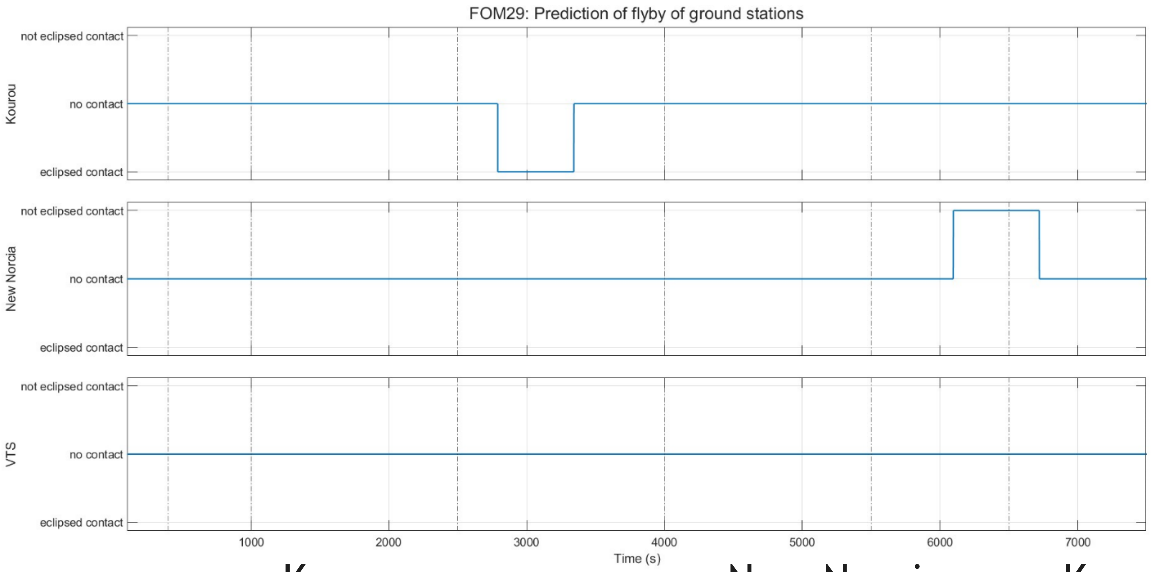
Test results

Ancillary services: flyby predictions

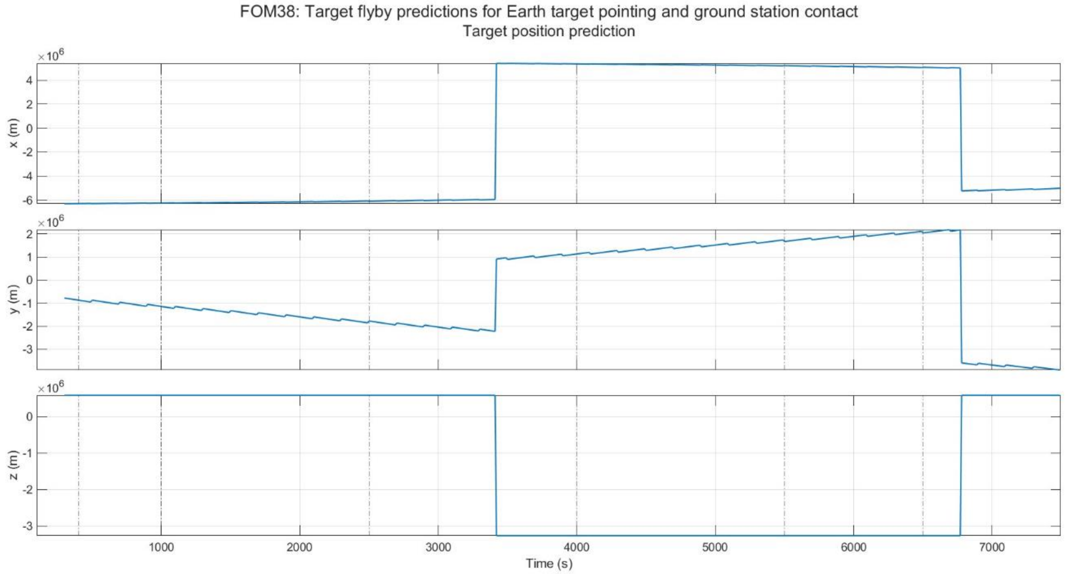
Flybys:

1. Kourou @~3000s for ~500s eclipsed
2. New Norcia @~6000s for ~500s illuminated
3. Kourou (predicted)



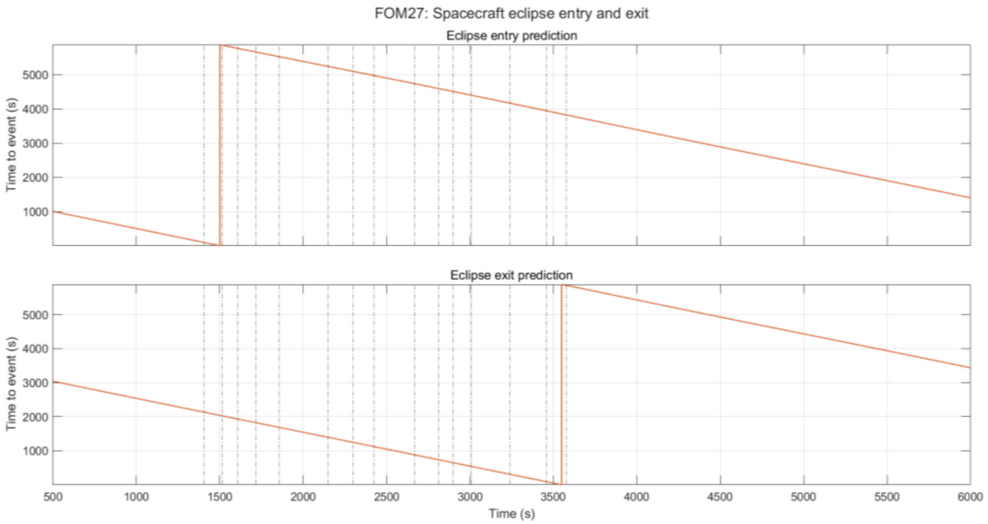


Kourou → ← New Norcia → ← Kourou



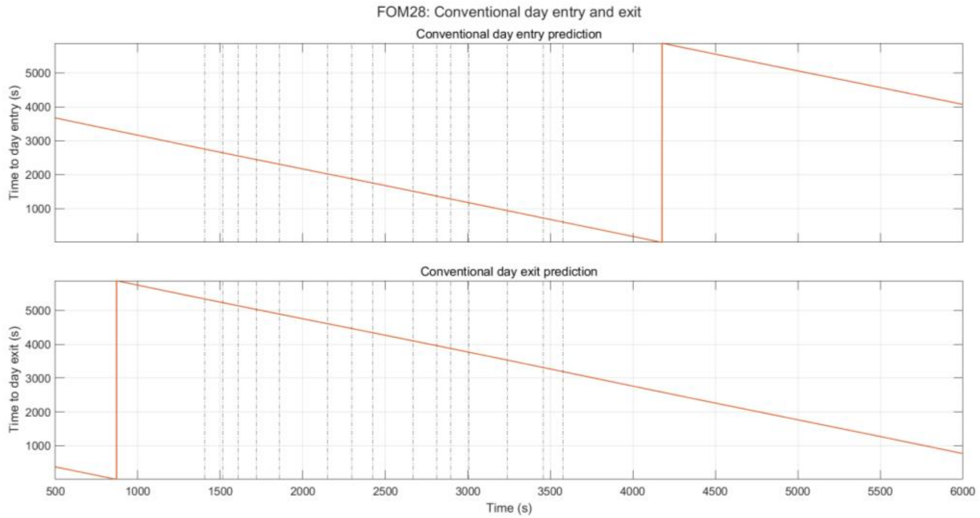
Test results

Ancillary services: crossing predictions


FOM27: Spacecraft eclipse entry and exit

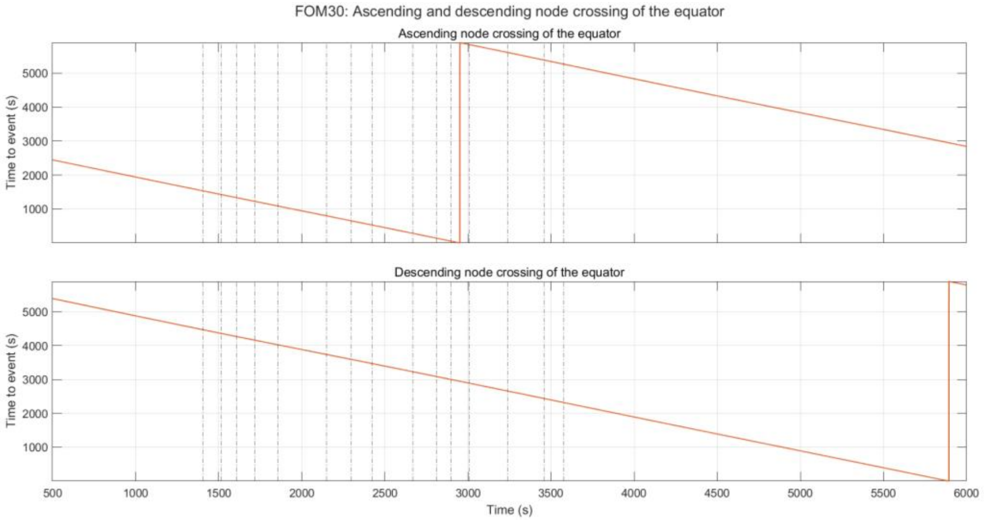
Eclipse entry prediction

Eclipse exit prediction


FOM28: Conventional day entry and exit

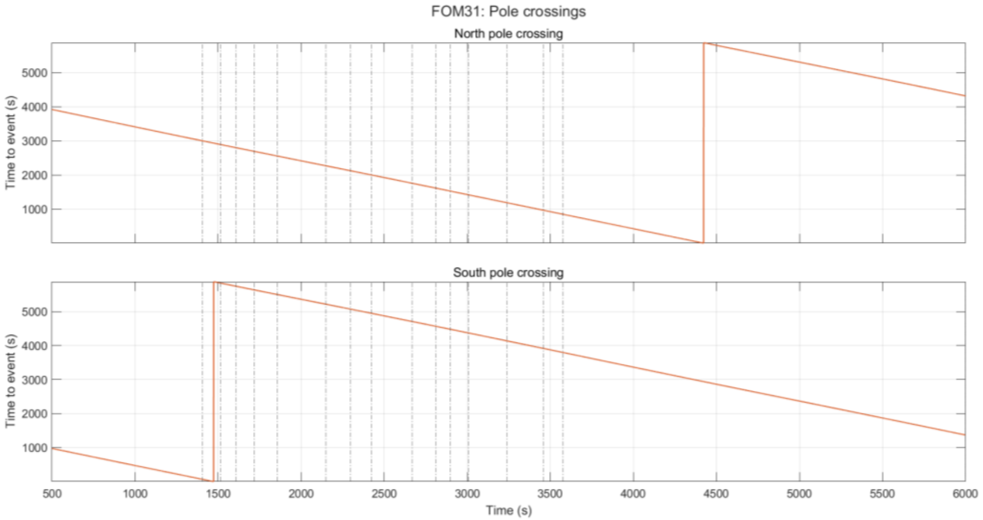
Conventional day entry prediction

Conventional day exit prediction


FOM30: Ascending and descending node crossing of the equator

Ascending node crossing of the equator

Descending node crossing of the equator

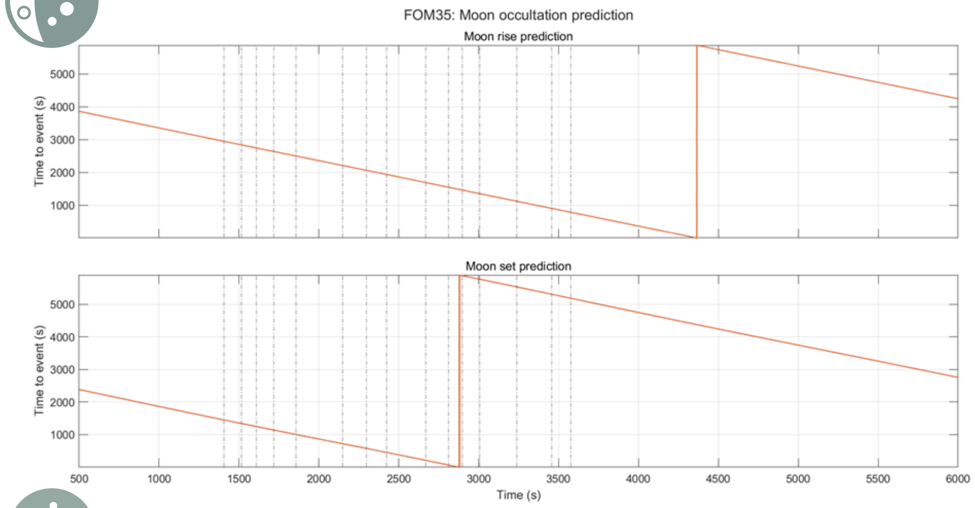

FOM31: Pole crossings

North pole crossing

South pole crossing

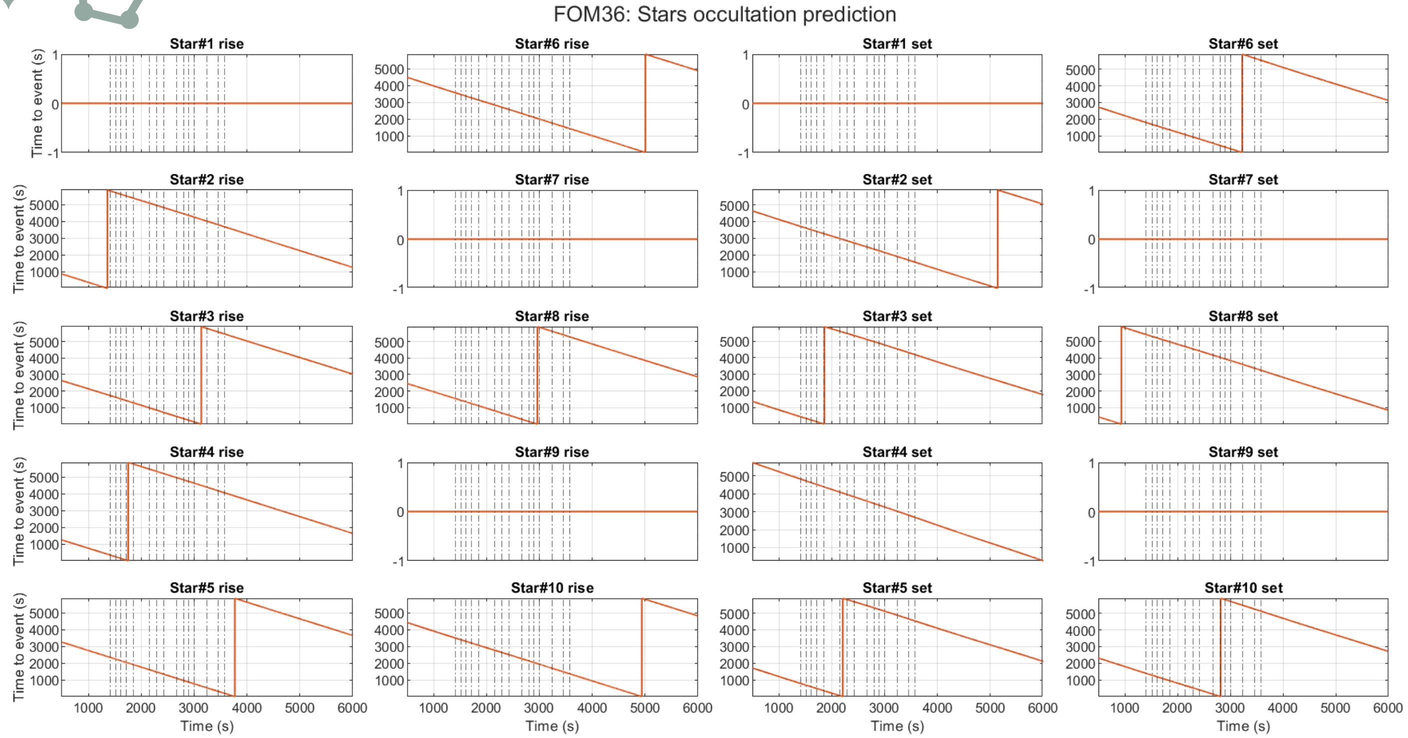
Test results

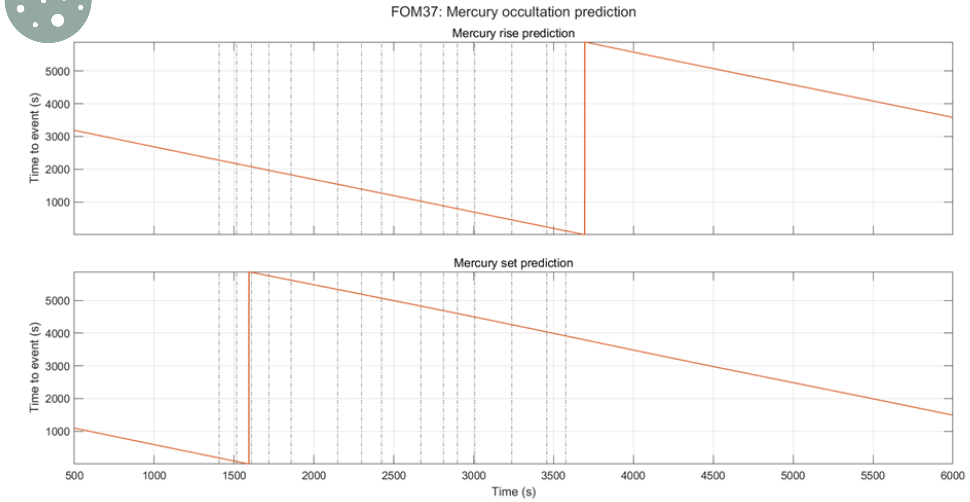
Ancillary services: planet/stars occultations


FOM35: Moon occultation prediction

Moon rise prediction

Moon set prediction


FOM36: Stars occultation prediction


FOM37: Mercury occultation prediction

Mercury rise prediction

Mercury set prediction

Summary

Conclusions

The single-mode AOCS architecture was proposed, designed, and extensively tested following a detailed V&V campaign.

Also, the results were compared with those obtained in the ALTIUS mission.

Several (potential) benefits in favour of a single-mode AOCS architecture were identified and confirmed (here we emphasize that further maturation steps are needed to fairly compare the results at a similar TRL). These can be summarised as follows:

1. A decrease in verification and testing effort is not really expected. However, if the developed units are further matured they can be reused in other missions and then a reduction in V&V effort is evident.
2. An enhanced HW modularity can be realized through the simplification of a replacement of one HW unit by another as the VS and VA abstract layers in the AOCS only need to be parametrically adapted to different unit types and configuration.
3. A reduction in the number of HW units may be realistic thanks to a different redundancy approach through the VS and VA concept.
4. A simplified (AOCS-level) FDIR design and verification is possible due to potentially less monitors and recovery actions (no mode dependencies). In addition, the (AOCS-level) FDIR can be much more versatile due to the ease of reconfiguration.
5. The overall operation of the AOCS SW can be simplified due to a smaller number of TCs, procedures, operational constraints.
6. The developed single-mode AOCS units can be easily reused in other missions of similar nature due to their genericity.
7. The spacecraft’s autonomy can be significantly increased.

Summary

Recommendations (1/2)

Despite these benefits, there are a few remaining open ends (potential research directions and challengers):

1. Generalize the concepts to enlarge the scope of application (i.e., fully covering generic AOCS/GNC SW needs). Examples are:
    a. Cover spin stabilization next to 3-axes stabilization
    b. Consider alternative/additional sensors and actuators
    c. Include interaction with payloads (as they may provide feedback on the SC attitude which can be included in the VS)
    d. Include aerodynamic drag/torques to control the attitude and reduce fuel consumption
2. Formulate a set of design and development guidelines for the other elements of the system (such as OBSW, ground operations, but also power/thermal/communication subsystems).

Summary

Recommendations (2/2)

3. Raise the TRL and mature the technology, especially focusing on
   a. enabling real-time embedded optimization for guidance, control, and actuator management. The possibilities are endless:
      i. Optimized guidance offers much potential in virtually any mission
      ii. The VA concept is also optimization-based and is already tested up to TRL 6
      iii. MPC-based control schemes may offer advantages over classical control approaches
      iv. On-board model or system identification (black/grey box)
      v. Incorporation of learning processes
   b. The VS is a promising navigation approach that is highly flexible and versatile. Further maturation would potentially yield a generic (prefab) solution that can be used in real missions with much reduced effort.
4. Apply the concepts to more general GNC applications. Many of the concepts are generic and can be applied in other applications too within as well as outside the realm of space applications. Examples are:
   a. science missions, launchers, re-entry vehicles,
   b. aircraft and UAVs.

sener REDWIRE

THANK YOU

www.aeroespacial.sener

in www.linkedin.com/company/sener

www.youtube.com/user/senerengineering