# GENEVIS

*Generic Vision-Based Navigation Technology Building Blocks for Space Applications*

## A full-software, Vision-Based Navigation solution dedicated to precise lunar landing

Paul DUTEIS, Roland BROCHARD, Sylvain TIBERIO and
Darius DJAFARI-ROUHANI (Airbus Defence and Space SAS)
Manuel SANCHEZ GESTIDO (ESA)

DEFENCE AND SPACE

November 2019

esa AIRBUS

# Agenda

***1- VBN Solutions for Rendezvous and Landing***

***2- Moon Landing Scenario: VBN Design***

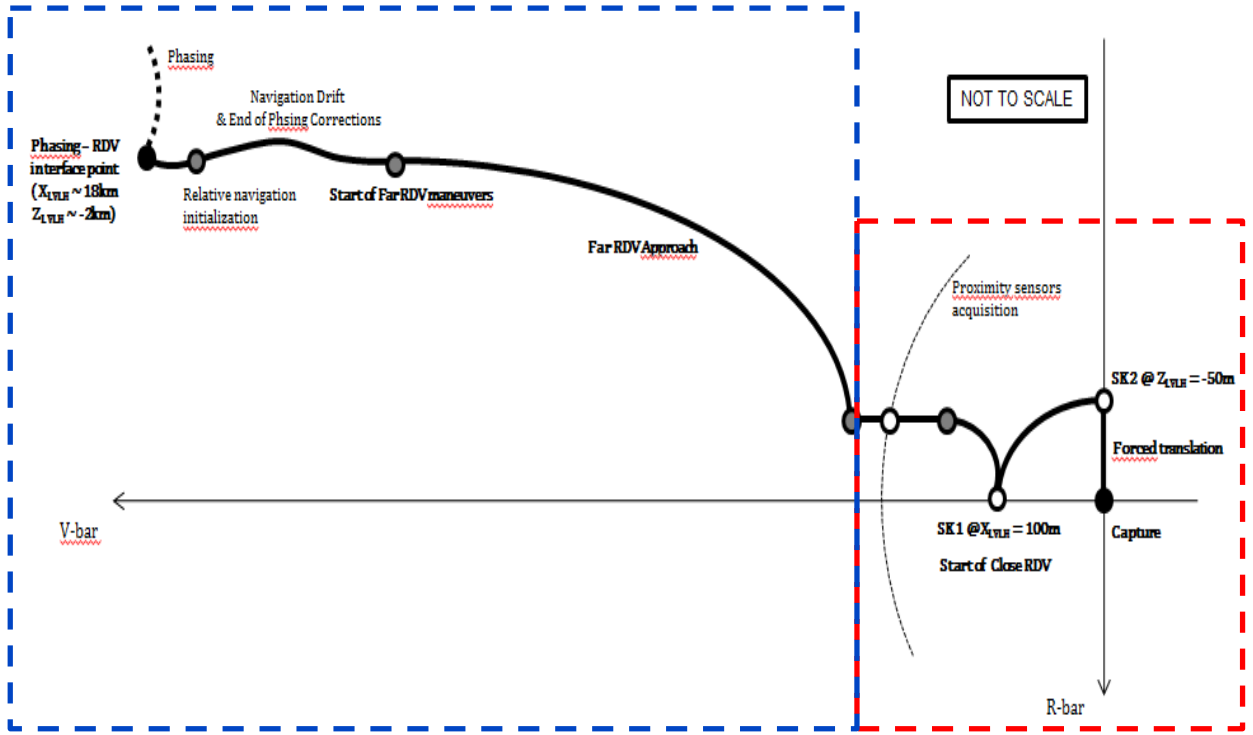
***3- Moon Landing Scenario: Validation Tests***

***3.1- Simulation Tests***
***3.2- Real-Time Tests with Space Processor in-the-loop (PIL)***
***3.2- Flight Tests with Representative Sensors / Hardware in-the-loop (HIL)***

# VBN Solutions for Rendezvous and Landing

# Typical VBN Scenarios for Rendezvous & Landing

## ❑ Rendezvous



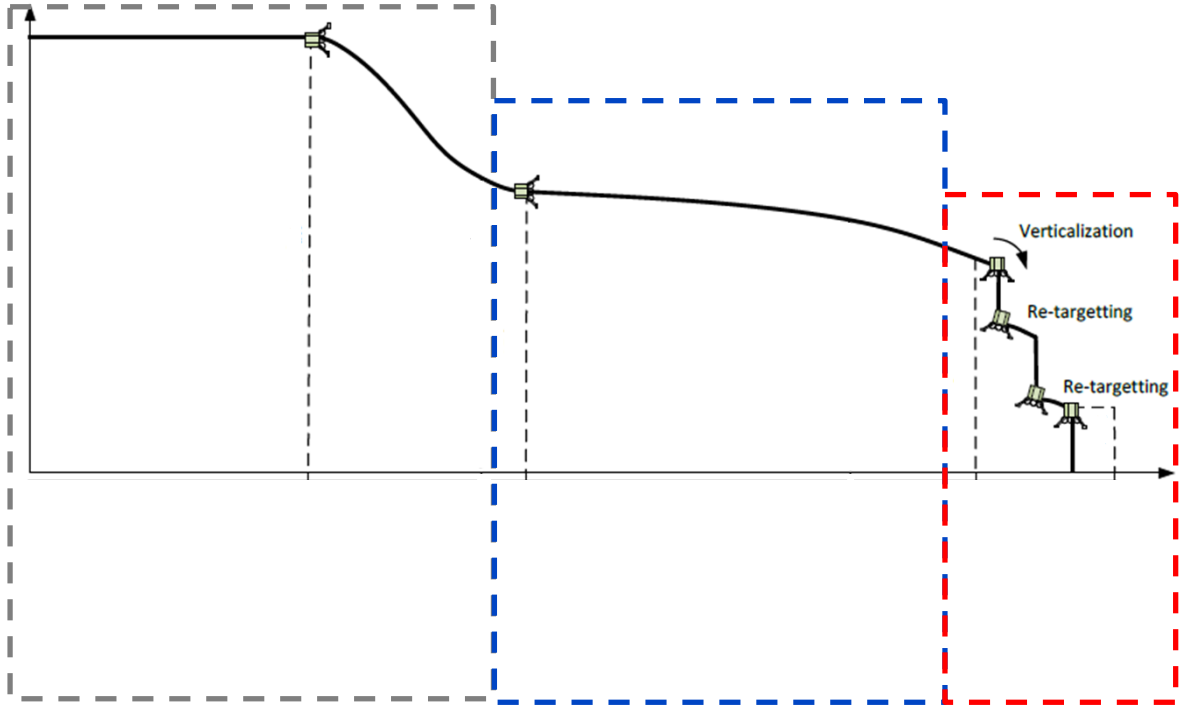

End of Phasing
& Far Rendezvous Operations

Centre of Brightness
2DoF

Close Rendezvous Operations

Model-Based Matching
3DoF or 6DoF
*(considered or estimated attitude)*

## ❑ Precision Landing




Observation
& Transfer Orbits

Ground Navigation (RF)
*(typically no VBN)*

Main Braking

Landmark Matching
3DoF or 6DoF
*(considered or estimated attitude)*

Feature Tracking
2DoF or 5DoF
*(typical, depends upon implementation)*

Precision Braking & Landing

Feature Tracking
2DoF or 5DoF
*(typical, depends upon implementation)*

# Typical Key Performance & Design Drivers

## ❑ Rendezvous

| Point of Interest | Long Range *(typically ~1km)* | Long Range to Close Range Transition (typically ~100m) | Docking |
|---|---|---|---|
| **Relative Position** | 1% of range | 1m | 2cm |
| **Relative Velocity** | 25cm/s *(cross)*<br>50cm/s *(downrange)* | 1cm/s | 0.1cm/s |
| **Relative Attitude** | N/A | 10° | 1° |

- ✓ Drivers for image processing: lightning conditions, sun avoidance, stray light minimisation and robustness to third bodies in the FoV
- ✓ Constraints by the capture mechanism: drives the final GNC performance
- ✓ Dynamics: very slow to slow (tumbling targets)

## ❑ Precision landing

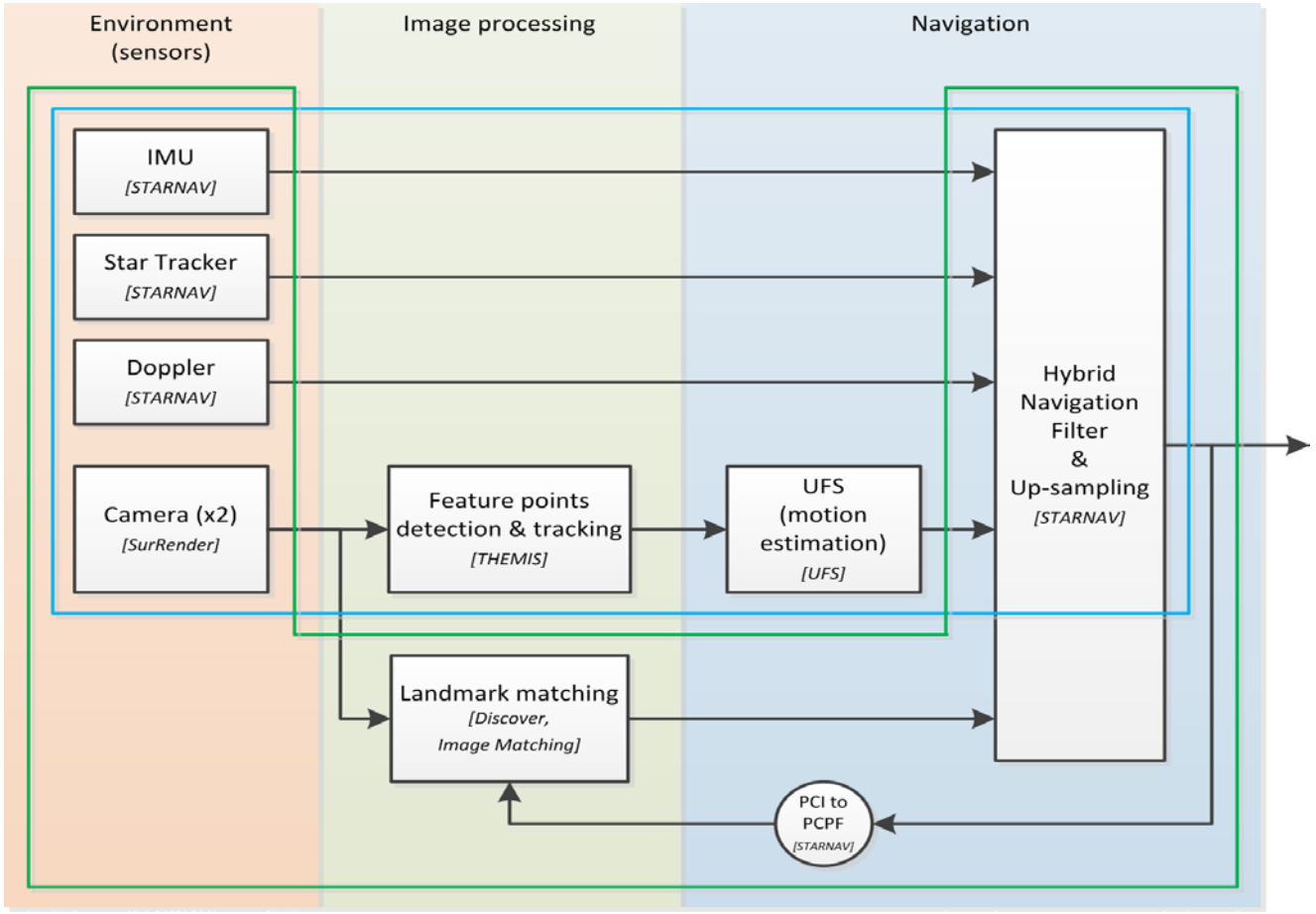
| Point of Interest | Main Braking | Precision Braking | Touchdown |
|---|---|---|---|
| **Position** | 200m | 200m<br>10m relative[1] | 2cm |
| **Velocity** | 2m/s | 10cm/s | 0.1cm/s |
| **Attitude** | N/A | 10° | 1° |

[1]*Performance relative to the designated landing site in the case of landing site retargeting (HDA using LIDAR, or other)*

- ✓ Drivers for image processing : lightning conditions, cast shadows, ground database availability
- ✓ Constraints by the hazards detection process: drives the navigation performance during precision braking
- ✓ Constraints by the landing gear: drives the final GNC performance
- ✓ Dynamics: medium to high

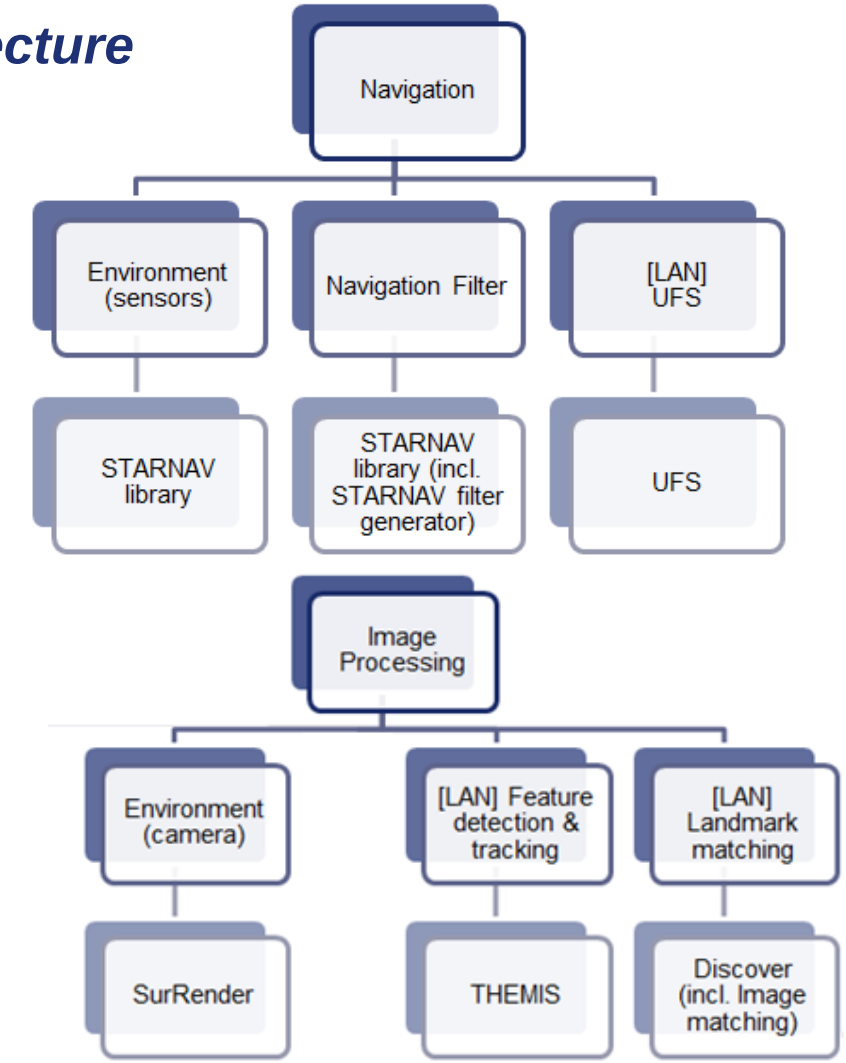
# Moon Landing Scenario: VBN Design

# Overview of the GENEVIS-LANDING Navigation Architecture




*GENEVIS-LANDING Architecture*

*Blue: relative navigation*
*Green: absolute navigation*

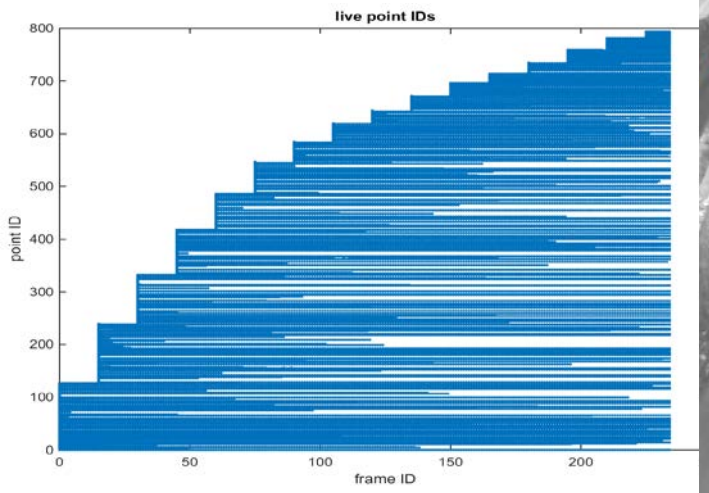


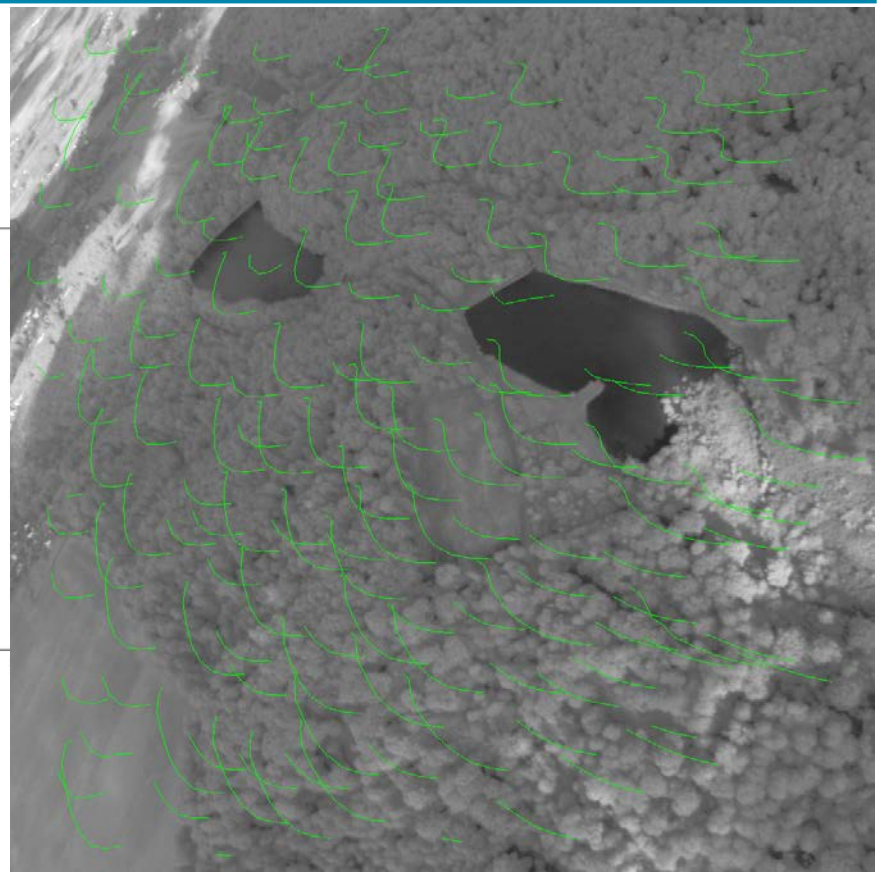

*Navigation and IP Components*

DEFENCE AND SPACE

# *Feature Tracking: THEMIS*

- ❑ ***Features***
  - ✓ Detected on the fly
  - ✓ Uniformly distributed over the image
  - ✓ Very fast and robust detection
- ❑ ***Tracking***
  - ✓ Enhanced KLT tracker
  - ✓ Points kept as long as possible to avoid random walk errors
  - ✓ Outlier filtering
  - ✓ Robust to rotations, scaling and large translations



➢ ***Detection + Tracking = 5Hz on LEON4 (1 core, integrated with landmark matching and navigation filter)***



esa AIRBUS

DEFENCE AND SPACE

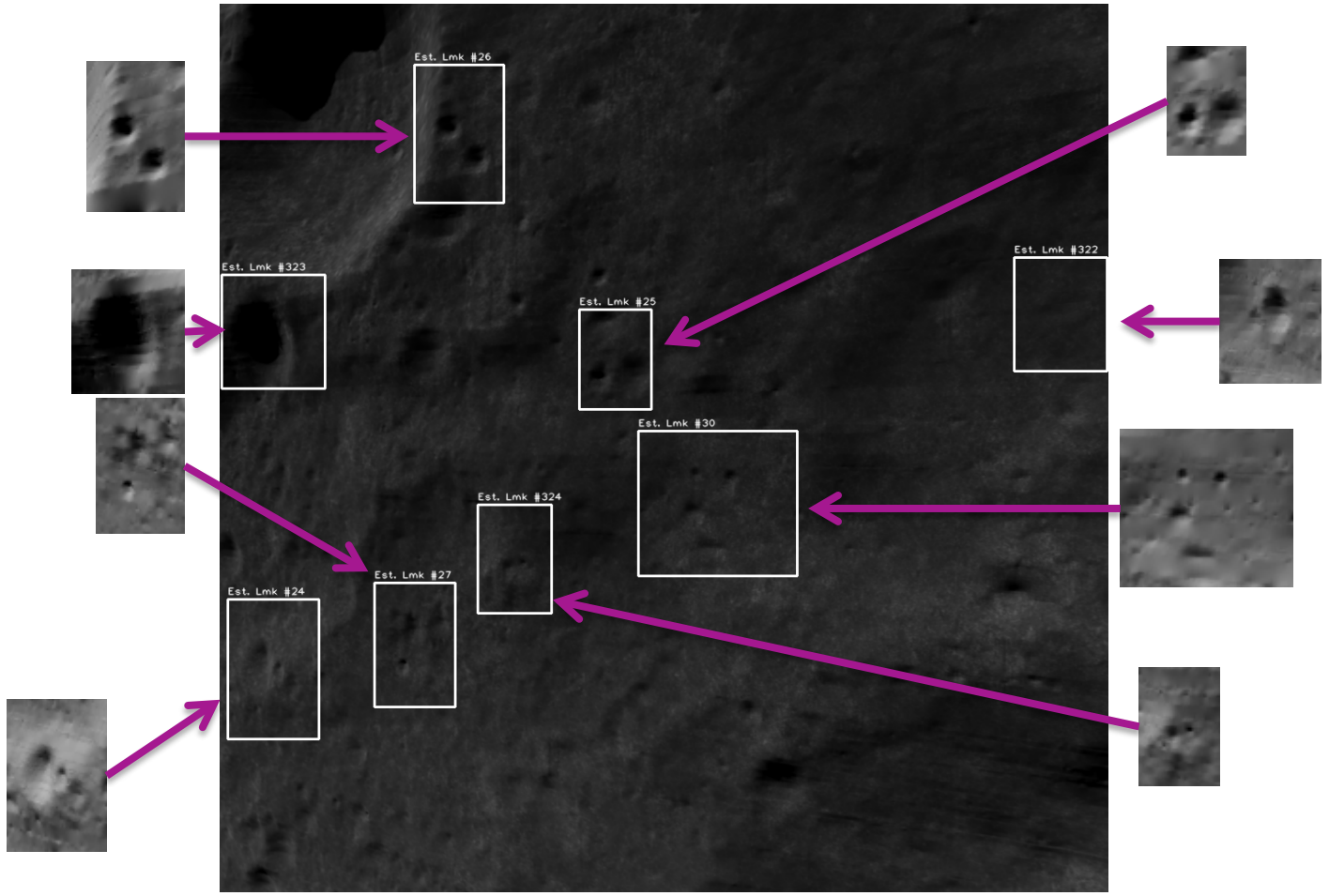
# *Landmark Matching: Discover*

## ❑ *Landmarks*

- ✓ Selected offline
- ✓ Recognizable features: craters, groups of craters, …
- ✓ Landmark views generated with raytracing
- ✓ Handles shadows and perspective effects

## ❑ *Matching*

- ✓ Zero Normalized Cross Correlation
- ✓ Each landmark is correlated with the whole image



# Navigation: Feature Tracking & Landmark Matching Preprocessors

## ❑ Preprocessor Role

The preprocessors:

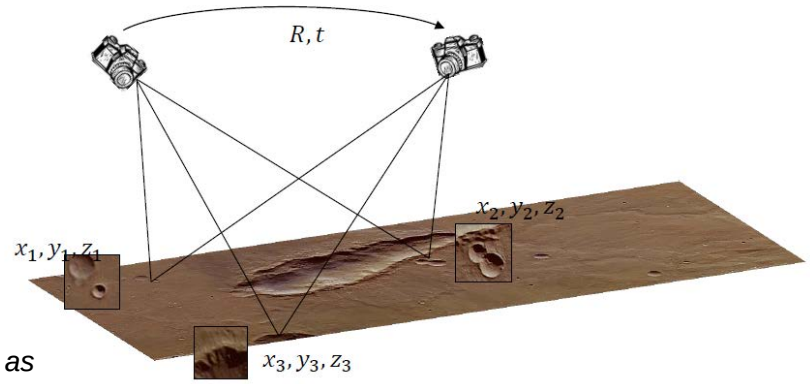
- ✓ Perform outlier rejection on the image processing outputs
- ✓ Aggregate the image-processing measurements in a condensed measurement that can be tied to the filter states
- ✓ Provide the information matrix
- ✓ Tackle the non-linearities due to vision prior to the EKF

## ❑ Landmark Matching Preprocessor

- ✓ Inputs:
  - I. (i,j) point coordinates in the image plane
  - II. Corresponding landmark positions in PCPF
- ✓ Output: 3DoF PCPF Position, by solving the PnP problem in closed-form (attitude aiding)

## ❑ Feature Tracking Preprocessor – UFS (Ultra Fast Slam)

- ✓ Inputs:
  - I. (i,j) point coordinates in the image plane
  - II. Corresponding point identifiers to track points across subsequent images
- ✓ Output: 2DoF Translation direction from key frame[1] A to key frame B

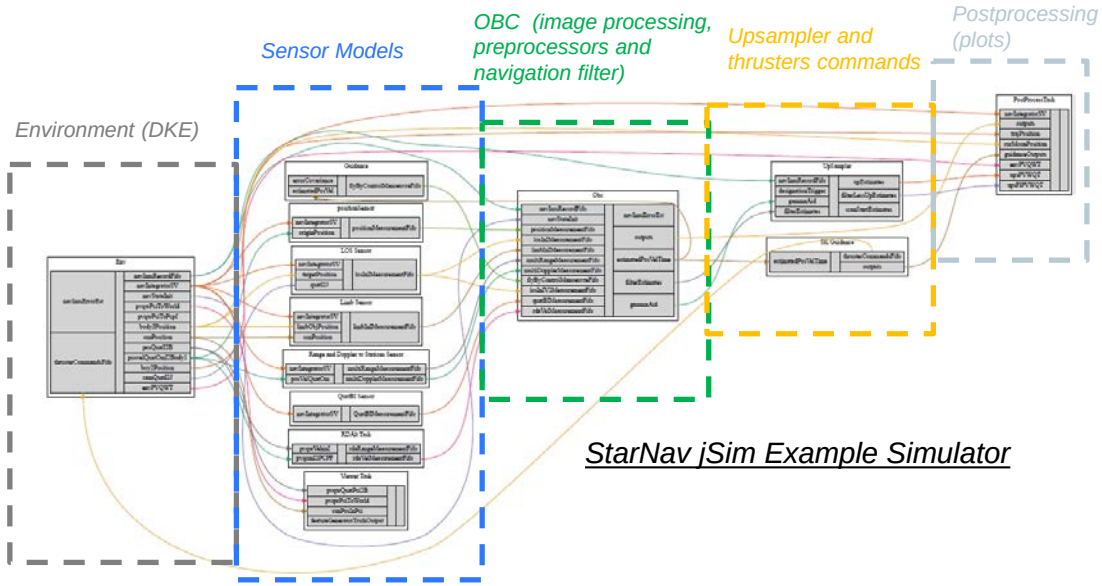



[1]*Key frames are triggered depending on relative dynamics with the objective of ensuring a long baseline (as long as point correspondences still exist) so as to reduce the impact of the tracker's noise on the measurement and maximise overall observability.*

# Navigation: StarNav Library & Schmidt EKF

## ❑ StarNav library

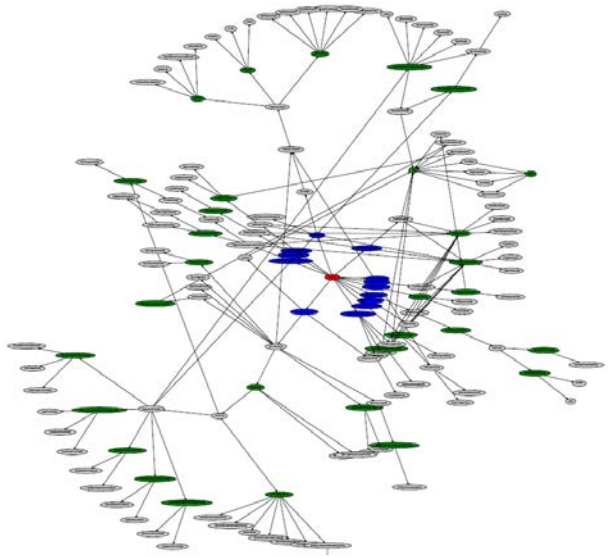
- ✓ In-house library for VBN and GNSS navigation, including sensor models, filter models, filter implementations and simulation means
- ✓ Code writing in Java, but with transcoding capabilities to generate C code respecting flight code standards




StarNav jSim Example Simulator

## ❑ Schmidt EKF

- ✓ The implemented EKF is a Schmidt (considered) EKF in UD form
- ✓ The propagation equations are generated to be able to be seamlessly connected to polymorphic sensors
- ✓ The covariance matrix is stored in UD form and its structure is accounted for (sparsity) to be able to stay compatible with the demanding requirements of embedded applications onboard satellites even with large state vectors (typically landing applications)



StarNav Model Polymorphism

# Moon Landing Scenario: Validation Tests

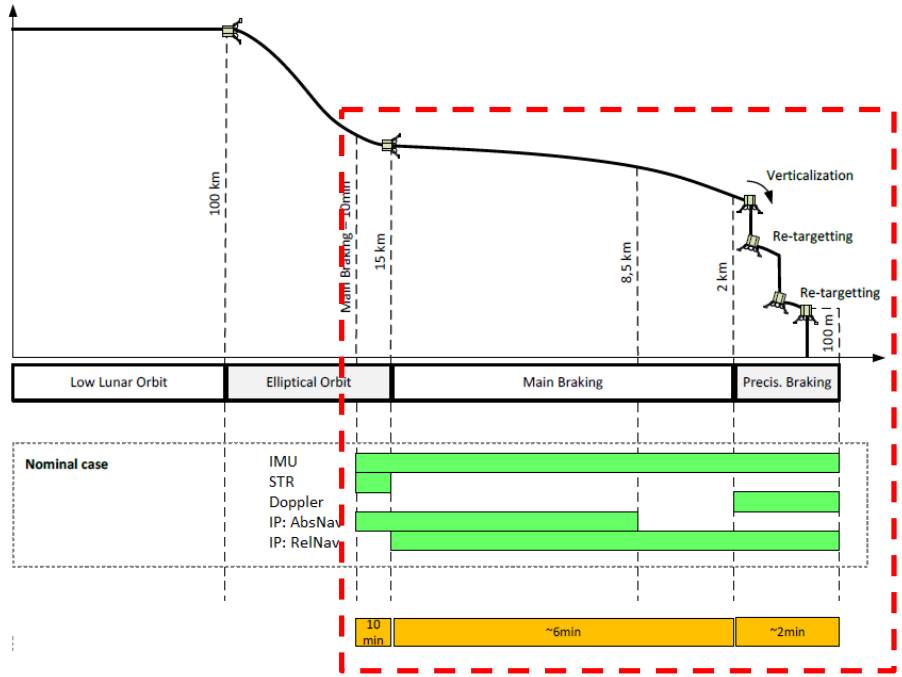
# Validation Models & Simulation Timeline

## ❑ With performance models

- ✓ Performance models for feature tracking and landmark matching are used to perform Monte Carlo simulations
- ✓ Designed thanks to a previous assessment of the error contributors in the functional algorithms using past heritage

## ❑ With functional IP algorithms

- ✓ A reduced number of runs are performed with the functional algorithms to check the behaviour is as expected

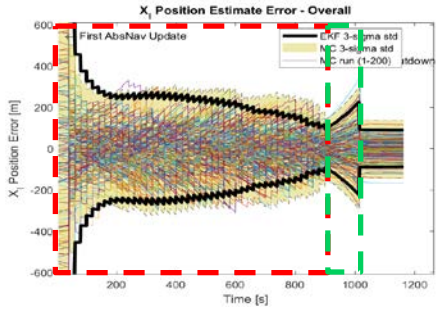


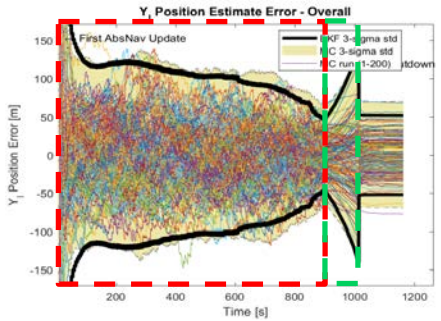

VBN Simulation Window

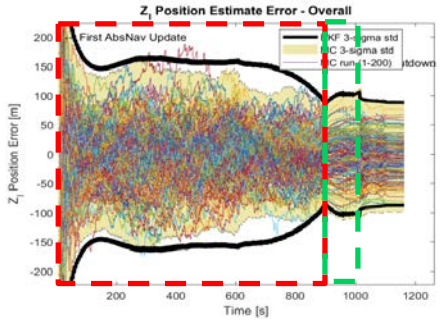
| Sensors & Pseudo-Sensors (IP algorithms) per Phase | MB (~15 - ~2km) | PB (~2 - 0km) |
|---|---|---|
| Doppler sensor | No | Yes |
| Inertial Measurement Unit | Yes | Yes |
| On-Board Gravity Model-Based Prediction | Yes | Yes |
| IP: Absolute Navigation | Yes until 8.5km (TBC) | No |
| IP: Relative Navigation | Yes | Yes |
| Wide FoV (~70°, 1024x1024, 5Hz) visible cameras | Yes | Yes |

| Scenario | Timeframe [sec] | Duration [sec] |
|---|---|---|
| Elliptical orbit exit phase (free-flying) | [0 ; 600] | 600 |
| Main Braking phase | [600 ; 1005] | 405 |
| Verticalization | [1005 ; 1020] | 15 |
| Precision Braking, including LS designation from t=1040s to 1060s | [1020 ; 1160] | 140 |
| Terminal Descent | [1160 ; 1165] | 5 |

# Results with Performance Models




*Position error [m], expressed in PCI frame, wrt time [sec]*

## ❑ Position error budget

- ✓ ≤300m along each PCI axis, 150sec after filter start

## ❑ (1) AbsNav window [0 ; 900]sec

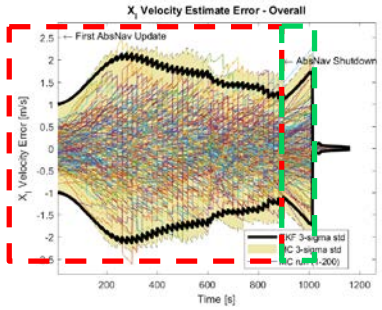
*Constant performance increase*

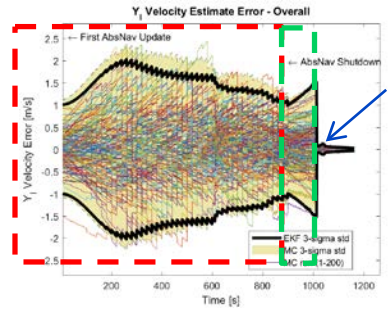
## ❑ (2) No AbsNav, at t>900sec

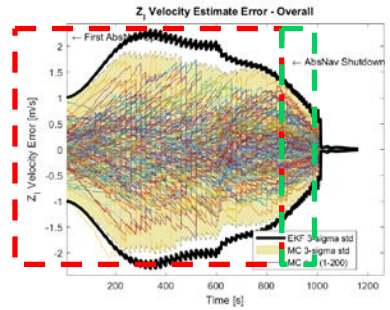
*Filter based on IMU, and RelNav only.*
*Drift in position performance*

## ❑ (3) No AbsNav, but RDA, at t>1000sec

*When the S/C is verticalized, RDA measurements are received.*
*Drift coming from the lack of knowledge in position after AbsNav shutdown is instantly and mostly recovered.*
*Stable position performance from this point on, thanks to RDA (providing good velocity knowledge).*




*Velocity error [m], expressed in PCI frame, wrt time [sec]*

## ❑ (1) AbsNav window [0 ; 900]sec

*Velocity first drifts away in performance: filter only has velocity information coming from IMU.*
*But after some time (~300sec), AbsNav position measurements provide enough to enhance pure IMU navigation.*

## ❑ (2) No AbsNav, at t>900sec

*Filter based on IMU, and RelNav only.*
*Drift in velocity, except along Z axis: RelNav provides more observability along this axis than along other ones.*

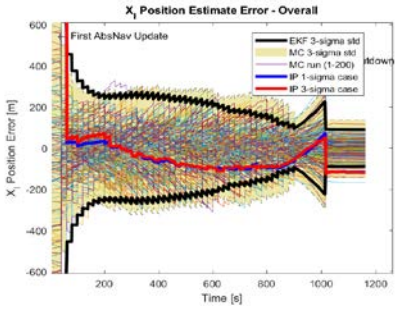
## ❑ (3) No AbsNav, but RDA, at t>1000sec

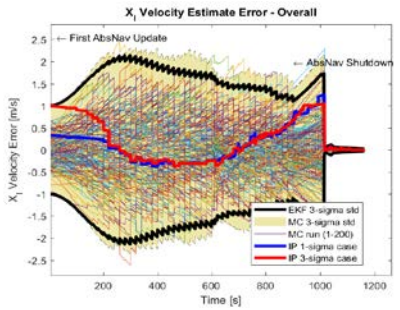
*When the S/C is verticalized, RDA measurements are received and leads to instantly very good velocity estimations.*
*Stable velocity performance from this point on, except during HDA scan (velocity manoeuvres) at (4)*

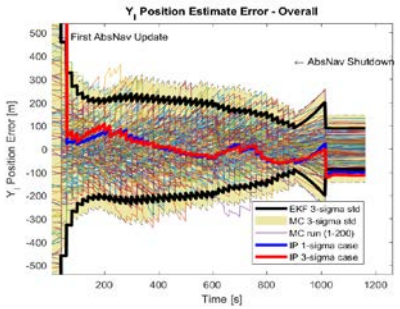
esa AIRBUS

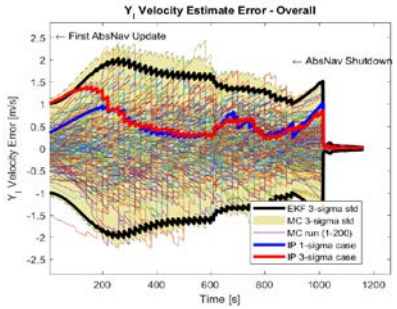
# Results with Functional IP & "No Vision" Comparison

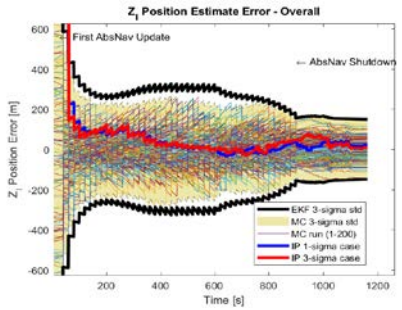
- ***Runs with IP models are consistent with Monte-Carlo 3σ envelope***

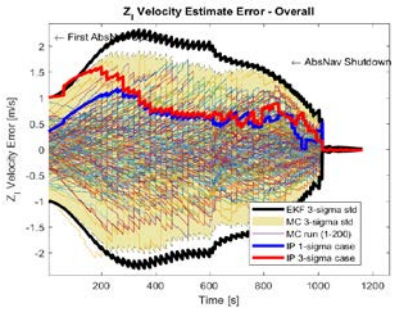


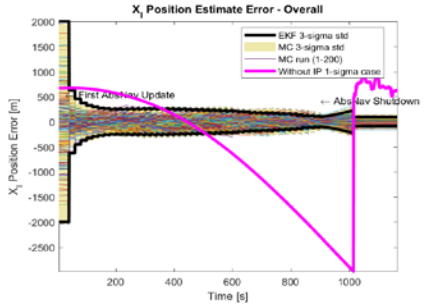



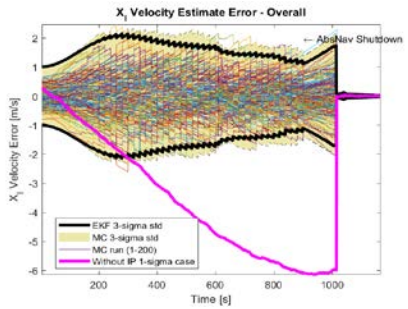



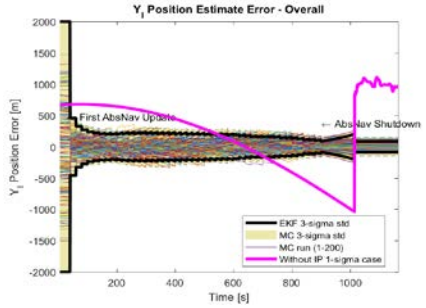



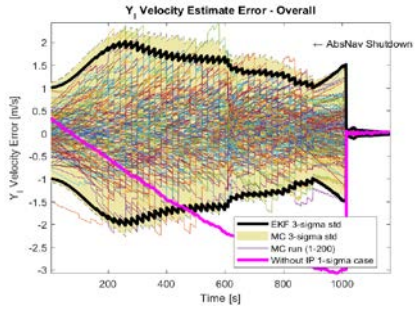



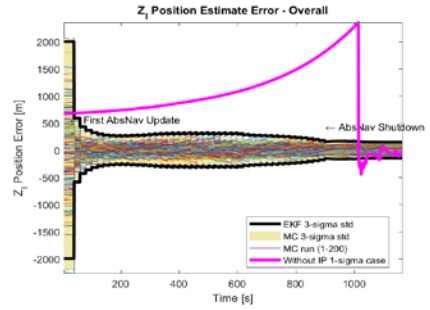

- ***Performance comparison with a « no vision » scenario***

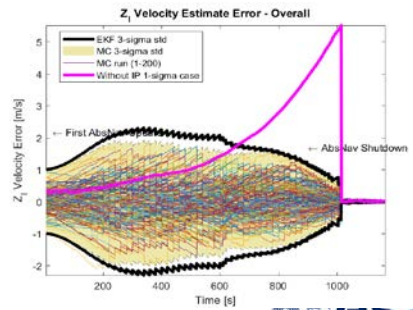



*PCI position error [m] and velocity error [m/s], wrt time [sec]*

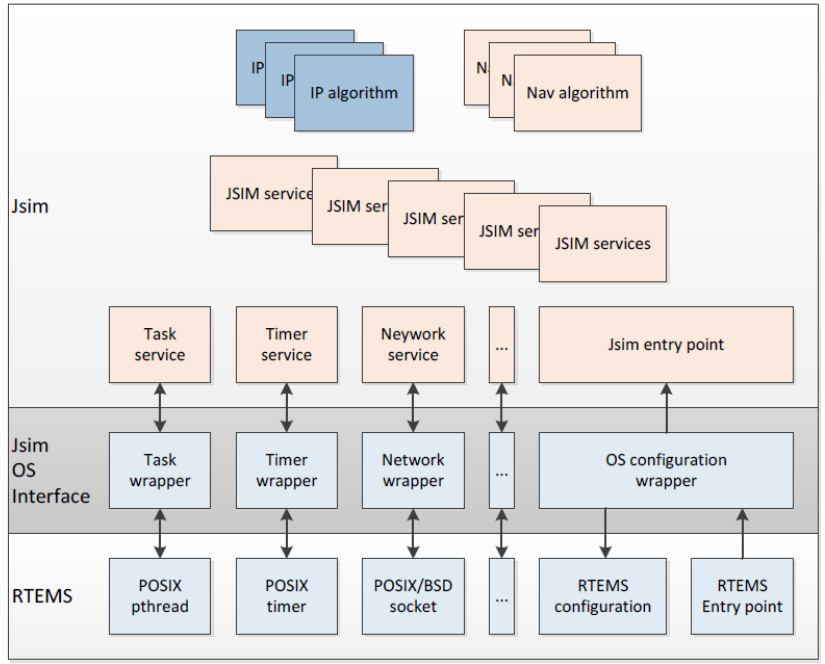
# Real-Time Simulation Architecture on GR740

## ❑ Porting algorithms on a GR740 running RTEMS

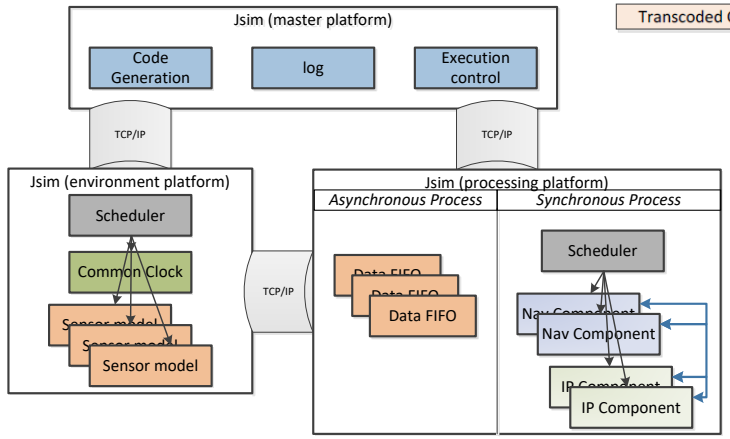
- ✓ C++ code for IP
- ✓ Transcoded C for navigation
- ✓ Native C for OS-level wrappers

## ❑ Scheduling logic

- ✓ Code generation, compilation and simulation control is performed on a master platform (PC)
- ✓ DKE and sensors are simulated in real-time on an “environment platform” (Linux PC)
- ✓ Sensor measurements are sent over Ethernet to asynchronous tasks awaiting data reception (simulating data transfers from actual sensors)
- ✓ Navigation and IP algorithms are scheduled synchronously in real-time on the processing platform (GR740 running RTEMS)




*Software Layers on GR740*




*Multi-platform Simulation using jSim*

# Profiling & Validation Strategy on GR740

## ❑ Profiled execution times

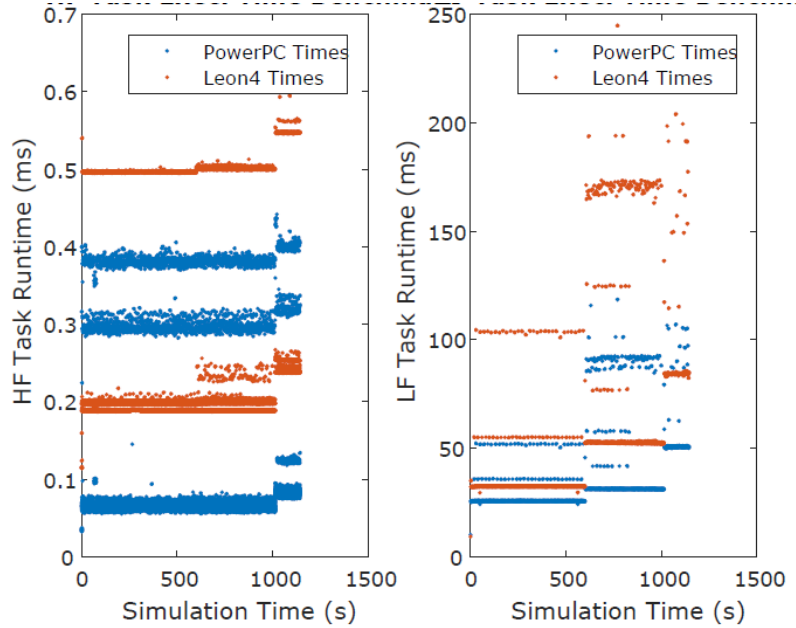
- ✓ The sum of the worst-case CPU loads taken separately for each component fit (just) on **one single core of the LEON4**
- ✓ Navigation has a variable execution time due to the particularity of processing feature tracking only on key frames

## ❑ Validation strategy on GR740

- ✓ Selected reference runs from previous simulations are replayed (same seeds) on the real-time, multi-platform architecture
- ✓ The objective is to demonstrate identical results (bit-wise) to validate the architecture (IOs occur identically, both in data and in sequence)
- ✓ Then, confidence in the design means most of the validation can be performed on PC simulation with no scope reduction

| Algorithms | Frequency [Hz] | Worst-case CPU load (on LEON4, 1 core) |
|---|---|---|
| IP: DISCOVER, landmark matching | 0.05 | 33% |
| IP: THEMIS, feature tracking | 5 | 35% |
| NAV: UFS + EKF | 1 (LF task)<br>20 (HF task) | 25% |

*Worst-Case CPU Loads*



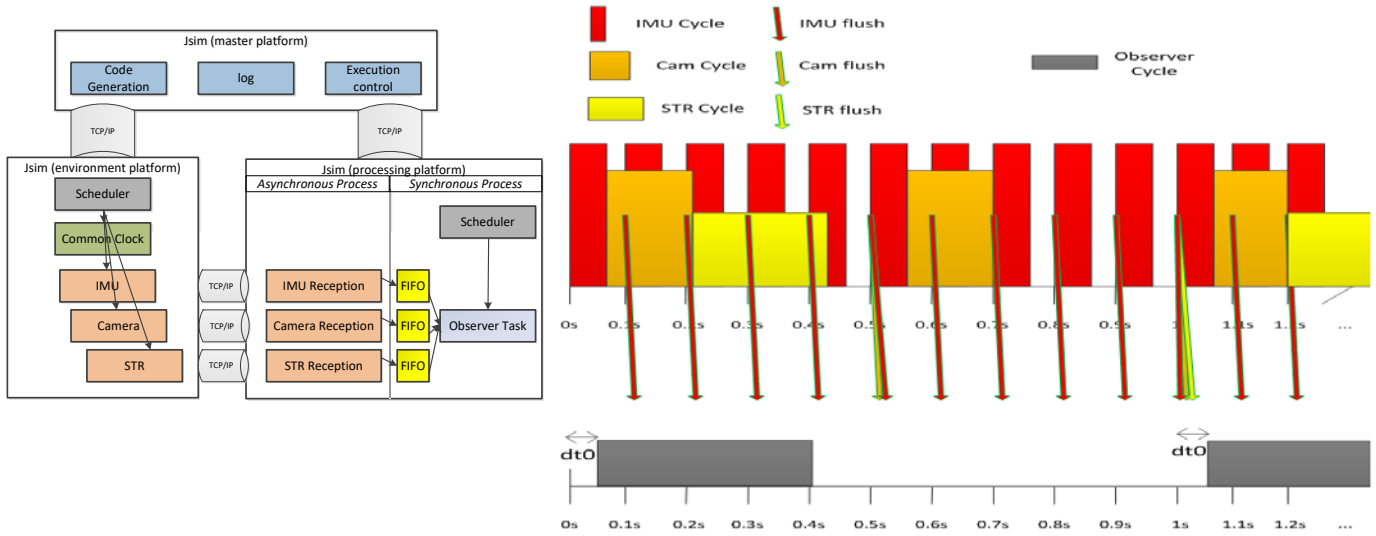

*CPU Execution Times: Navigation Filter*

DEFENCE AND SPACE

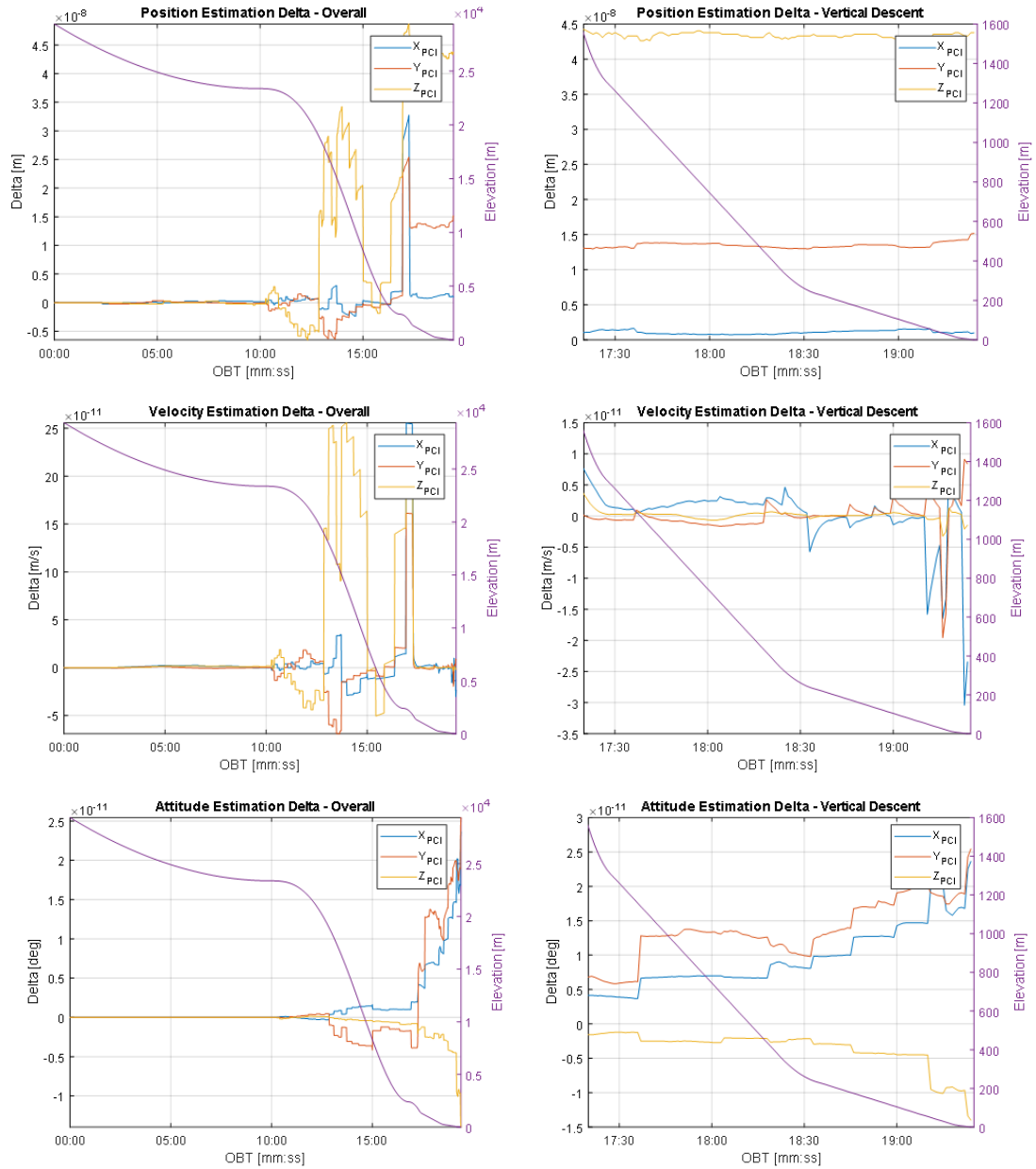
# Test Results on GR740

## ❑ Profiled execution times

- ✓ The reproducibility of the results have been ensured up to numerical errors on the studied reference scenarios
- ✓ Bit-wise reproducibility was not reached due to compiler specificities for floating operations on the GR740 platform but could be reached with additional efforts




*Ensuring I/O consistency between simulation and real-time, multi-platform on a simple example*




*Output difference between simulation (PC) and real-time, multi-platform*

# Flight Test Platform: Architecture Overview

## ❑ Sensors

- ✓ 2 IDS NIR cameras (1024x1024, 8bits)
- ✓ Sensonor STIM300 IMU
- ✓ North RtKite + Trimble GNSS receivers: to provide a reference trajectory

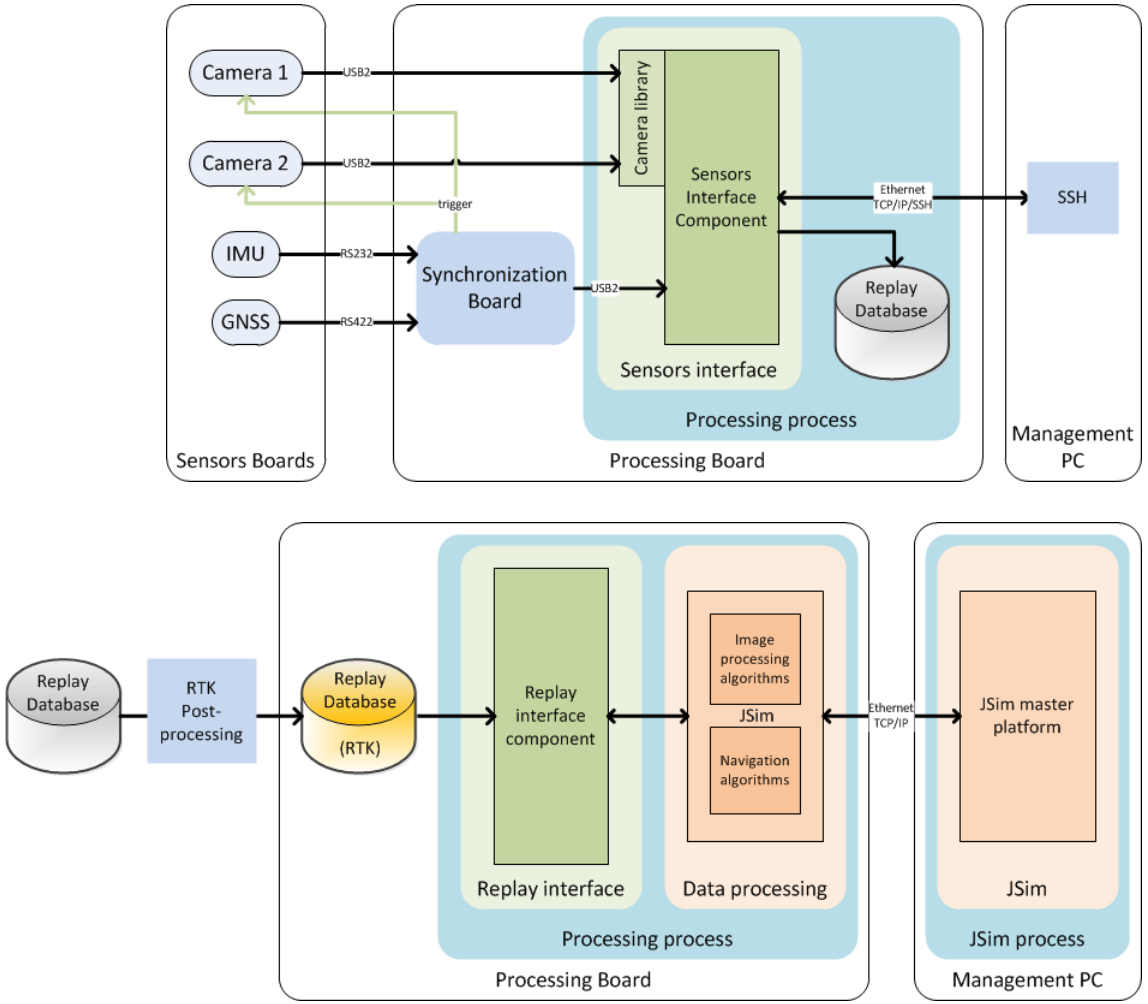
## ❑ Data acquisition

- ✓ PIC32 Microcontroller board to timestamp all data with a unique clock with high precision using HW interrupts
- ✓ Sensor data is recorded together with information on reception delays and timestamps

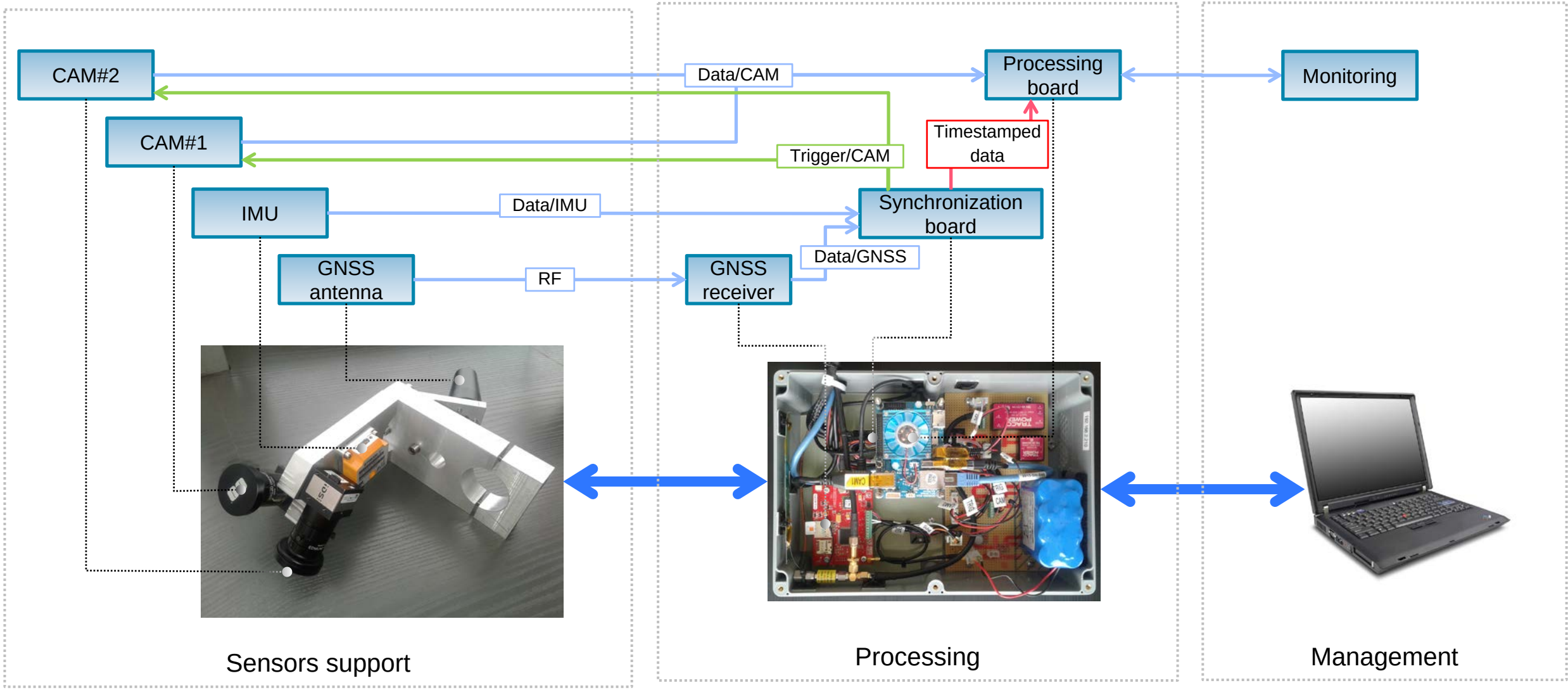
## ❑ Post-Flight analysis

- ✓ GNSS raw RINEX data is post-processed with known references to obtain a RTK-precise position reference
- ✓ Replay is possible and respects rigorously the live acquisition timeline and data[1]

*[1]Real-Time, live runs of the algorithms during flight is also possible, but requires NCOR connections to retrieve live RTK corrections and this proved difficult during helicopter flights.*



# Flight Test Platform: Components and Connections






esa AIRBUS

DEFENCE AND SPACE

# *Flight Tests Onboard Cabri G2*

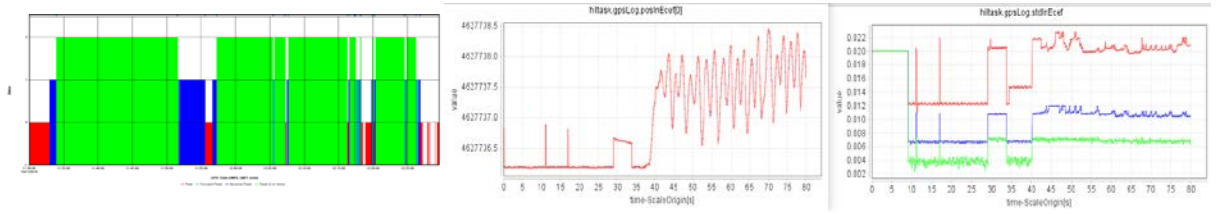
## ❑ *Objectives*

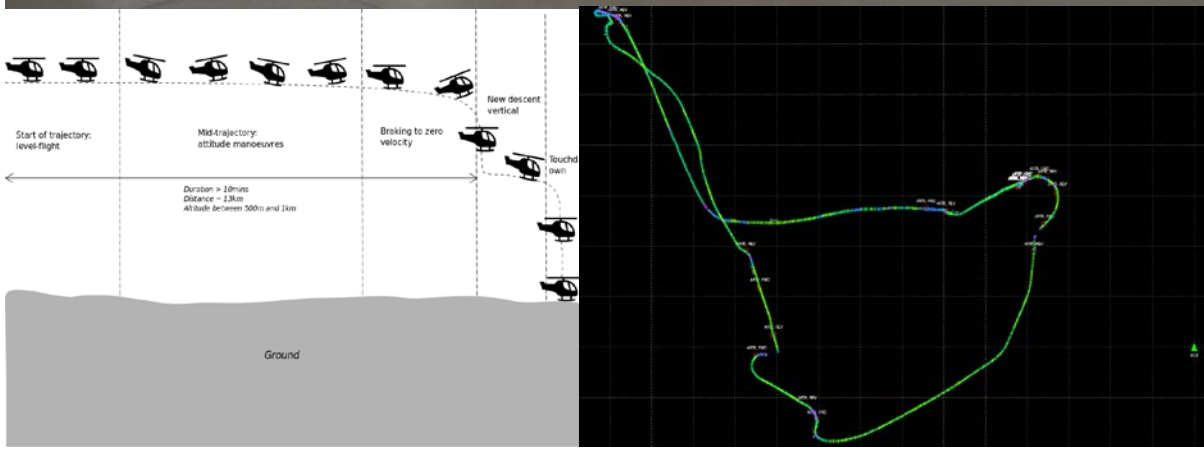
- ✓ The simulation tests demonstrated the performance level of the algorithms
- ✓ The GR740 tests demonstrated the real-time capabilities of the algorithms on a space grade architecture
- ✓ The flight tests demonstrate the correct behaviour of the algorithms with real data and representative dynamics, and validate our models

## ❑ *Scenario*

- ✓ Onboard a Cabri G2 in Toussus-le-Noble (Paris)
- ✓ Trajectory representative of a lunar landing: long flight (difficult on a drone), and final landing simulated in autorotation for higher vertical speeds

## ❑ *Data analysis currently on-going!*






*Flight Trajectory Overview (10/10/2019)*

# Conclusions

# Conclusions

- ***Technology demonstrator***
  - ✓ Successful demonstration of the estimation performance and real-time capabilities in a **full software** design
  - ✓ TRL5 reached with GENEVIS thanks to the demonstration on both Real-Time Test Benches (RTTB): the “Space RTTB” and “Flying RTTB” on a helicopter

- ***Technology with high genericity***
  - ✓ No strong assumptions on the availability of sensor data (fully asynchronous)
  - ✓ No strong assumptions on the terrain’s structure for absolute navigation (e.g. not necessarily craters)
  - ✓ Reconfigurable navigation filter (depending on the choices for the sensor suite, estimated states…)

  - ➢ The demonstrated precision landing is applicable to other types of bodies/planets and ground structures
  - ➢ This Vision-Based Navigation solution can be used to land on the Moon, on asteroids, on Mars…

- ***Future developments***
  - ✓ Demonstrate the final landing phase is possible using only vision (no Doppler/altimeter) by observing the scale factor ambiguity using the known braking force.

# Thank you



AIRBUS