DOI: 10.13009/EUCASS2019-38

Design, Development, Validation and Verification of GNC technologies

*A. Pellacani * and M.Graziano*, M. Suatoni**
**GMV aerospace and Defence*
Isaac Newton 11 PTM, 28760 Tres Cantos, Madrid

Abstract

Space missions are becoming more and more challenging, requiring fast and adaptive reactions to external stimuli that can be obtained with higher level of on-board autonomy. Because of such requirements, Guidance Navigation and Control technologies are increasing in complexity and, as a consequence, demand a more structured validation plan, to be started already during the preliminary stages of the mission.

GMV standard for the Design, Development, Validation and Verification of GNC technologies is an incremental validation that starts with unitary testing of functions and modes up to flight code run inside representative processors and with engineering/qualification sensor models in close loop.

The critical review of requirements allows for the derivation of GNC subsystem specification and for a preliminary design of GNC modes and functions necessary to achieve mission objectives. First step is functions unitary testing, but using previous experience and heritage, it is possible to test the algorithms in a representative closed loop GNC prototype already in phase A-B1. Model-In-the-Loop simulations demonstrate the feasibility of the preliminary design and the robustness of the selected solutions using Monte Carlo test campaigns. Being able to perform such tests during the preliminary stages of the development allows for efficient iterations at system level, giving valuable contributions for trade-offs that involve other subsystems.

A fast prototyping based on autocoding is performed as following step, verifying the GNC code in Software-In-the-Loop tests. To demonstrate the feasibility of the design, the Processor-In-the-Loop step is then a key point of the incremental validation in order to verify the behaviour of the GNC code in a representative Hardware and to identify the computational resources required through code profiling. At this stage it is possible that specific function/algorithm (e.g. Image Processing) turns out to be too demanding for a space qualified processor like a LEON. In this case, a HW-SW co-design is performed in order to get advantage of the HW acceleration that an FPGA can offer. An estimation of the required resources and HW units is an essential input for Spacecraft system in order to perform budget activities and interfaces between subsystems. Furthermore, this high level of simulations fidelity, allows to take into account also operations constraints, defining a detailed time line of the different mission phases.

In the final stage of the incremental validation Hardware-In-the-Loop is performed, including representative sensors in the loop (e.g. camera in a vision-based GNC). Two environments are foreseen for the HIL tests: an optical test bench where the images are generated on a high resolution screen and only a picture of it is taken; a robotic test bench (***platform-art©***) that re-creates a space-like scenario in terms of illumination conditions and simulated gravity using a dedicated laboratory with robotic arms. The ***platform-art©*** tests are the most complex to set-up and to execute, but there is a clear advantage in terms of representativeness because no artificial image is generated and the camera directly takes a picture of the target mock-up.

This paper will include the details on GMV DDVV approach, which allows to perform end-to-end GNC simulations at very early phases of the mission to bring into considerations aspects normally not really considered in detail till later mission phases. GMV standard allows (through very fast iteration loops and full algorithms/SW coherence approach) algorithms updates/iteration till phases where they should be nominally frozen (thus, gaining flexibility and introduction of last techniques in the late phases of the mission.

DOI: 10.13009/EUCASS2019-38

1. Introduction

The paper objective is to describe the GMV approach for the Design Development Validation and Verification of a GNC Subsystem, including high level strategy and GNC Application Software (GNC ASW). The "V-cycle" approach should be followed along the entire mission, from phase A to E, but the model-based rapid prototyping allows for an early implementation, which can spot issues in the architecture/solution during the preliminary phases of the project. The sooner the issue is identified, the cheaper its solution will be, both in terms of costs and time/Delay.

2. DDVV GMV approach

The GMV approach for the design, development, validation and verification strategy is based on autocoding of the GNC Matlab/Simulink models in order to generate ANSI C-code optimized for embedded systems. This development strategy is part of an integrated, coherent and incremental DDVV approach based on the chain:

FES/MIL ► Autocoding ► SIL ► PIL ► HIL

By FES/MIL it is meant a Functional Engineering Simulator used to test the GNC prototype in Model-In-the-Loop configuration, using only Matlab/Simulink. The autocoding of the GNC prototype will allow for Software-In-the-Loop (SIL) tests and when the GNC ASW is embedded into a representative processor (e.g. LEON) the Processor-In-the-Loop tests can be performed. The final step is the Hardware-In-the-Loop test campaign, in which for vision based GNC usually the camera is included in the GNC loop.

This autocoding chain can provide invaluable support during the Design and Development phases and possibility to test V&V requirements already at early and intermediate design phases, allowing fast design iterations and feedback and the possibility to correct design problems, thus minimizing the required effort. Furthermore, this approach will provide an additional flexibility in the GNC design and the associated iterations with the industrial team and the Agency. Detailed autocoding rules have been agreed with ESA in the frame of the AUTOCOGEQ project, in which also an autocoding wizard tool in charge of supporting the user in all the phases of the autocoding process has been developed and it's been used at GMV in all the GNC prototypes.

The DDVV tasks are interleaved all along the GNC activities and different partners are usually involved in different steps of this chain. It is therefore mandatory to establish a common DDVV approach to be applicable to all team partners and to be applied commonly all through the different tasks and technical phases already from the early activity phases. By following the stated DDVV approach, efficient reuse of all work and models developed during early activity tasks can be applied to later coming tasks (e.g. smooth integration of models in the MIL, smooth auto-coding...).

It shall be noticed that with this approach the GNC will be designed in Matlab/Simulink, using the Control Design and Simulation Environment (FES) and model driven software engineering. This Matlab/Simulink implementation will evolve in flight code using the autocoding tool chain. The autocoding development strategy consists in generating the GNC ASW C-code in a straightforward way, directly from the Simulink model of the GNC. Nevertheless, the compliance with ESA S/W development and quality standards shall be maintained; for this reason the specific impacts on the S/W development process related to autocoding use will be documented and justified.

Figure 1 depicts the V-cycle workflow between GNC and FDIR design and prototyping and the GNC specification, development and testing in SIL, PIL, and HIL environment, with the main tools and environments involved.

The GNC and FDIR DDVV will be based on the following steps:

- **Analysis of scenario and derivation of GNC/AOCS Software Specification**: The process of derivation follows the classical top down approach. The requirements are usually grouped in the SRD and traceability matrices are issued to maintain the link from the source of each requirement through its decomposition to implementation and verification.
- **Set-up of a Control Set-up and Simulation Environment (FES)** with the use of reference models of the selected algorithms and solutions for the GNC system, allowing having an environment to define, analyse and maintain the S/W lifecycle. This represents the main conductive design supporting tool and verification at algorithm level (Model in the Loop, MIL) all along the activity. Together with the FES design environment, it is fundamental to define modelling rules/guidelines for compatibility with the autocoding tools.
- **Autocoding of FES-validated GNC syste**m: the GNC ASW C code is generated and the S/W V&V process is started. In this environment static and dynamic verification of the C-code functions can be performed using LDRA tool by embedding the generated C-code of the ASW functions in the FES simulator as s-functions (SIL). Unitary and Integration tests are realised.
- **Preliminary GNC ASW validation environments**: preliminary functional validation tests (main ASW requirements verification versus technical specifications) are performed in SIL environment, by integrating the produced ASW C-code in the FES simulator and test it in closed loop, in order to provide potential feedbacks on the design already at early stage.

DOI: 10.13009/EUCASS2019-38

- **Validation and Verification in the SVF and PIL**: the SVF is a S/W based verification setup running the OBSW image in-side a processor emulator in no real-time conditions and fully simulated closed-loop environment. It includes also the autocoded models of the FES Real World (DKE, actuators, sensors, etc.). It allows SW integration between the GNC ASW and the OBSW in the OBC, to be tested in a PIL architecture; the formal functional validation tests (GNC ASW requirements verification versus technical specifications) are also performed.
- **Validation and Verification in the HIL facility (HILF)**: the GNC ASW, integrated in the OBSW, is then validated in the HILF, which includes the functional model of the OBC and the GNC SCOE, running in real time and closed loop environment. This facility is used for testing the correct integration of the GNC ASW and OBSW in the on-board computer and real time conditions and verification of the GNC ASW requirements. It is possible to also include qualification/engineering models of on-board sensors in the closed loop, like a camera. The camera would take pictures of the object projected on a screen (optical lab) or of a mock-up (***platform-art©***). The HILF testing will be considered the last step in order to reach the GNC Software qualification status.

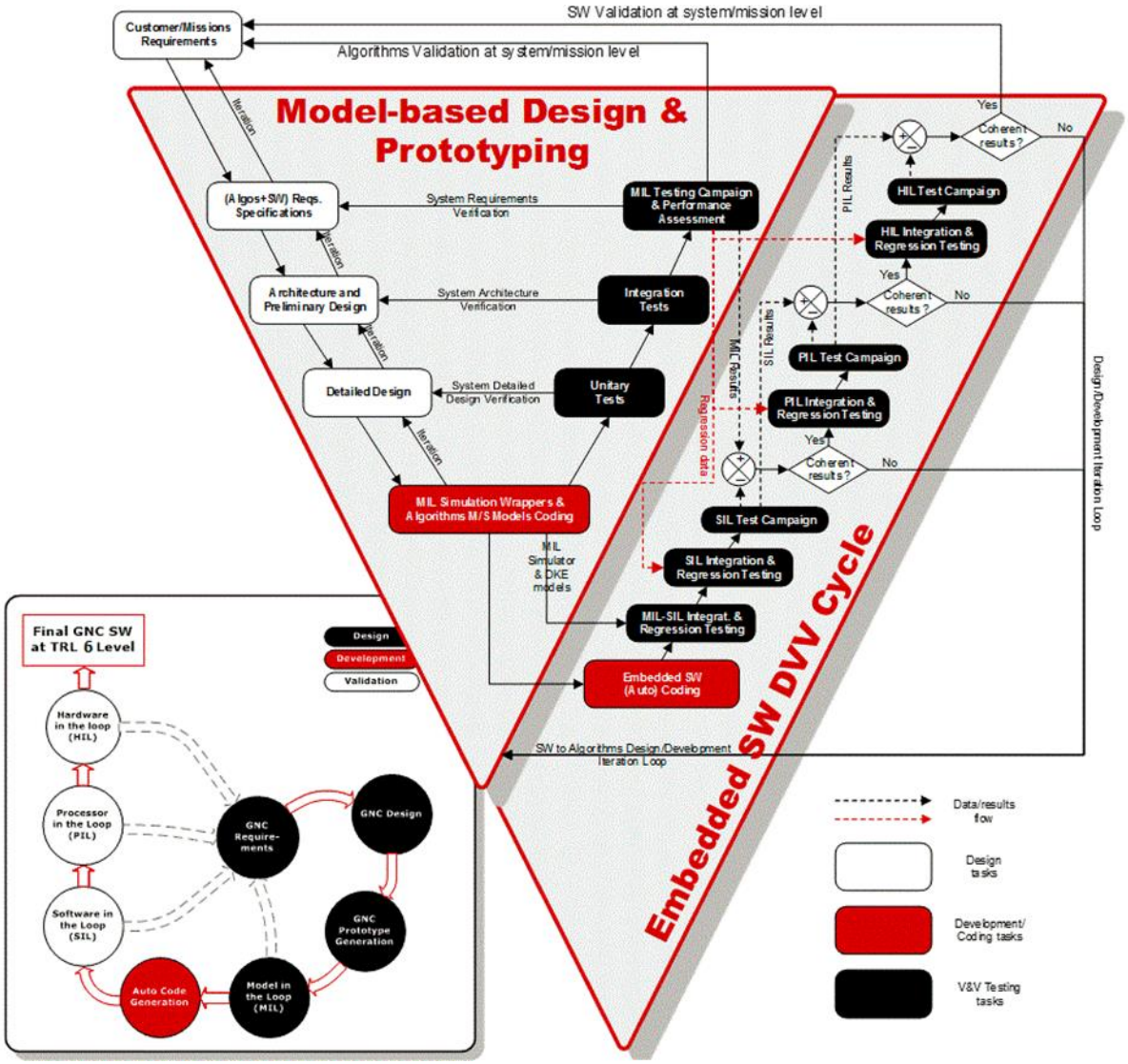

Figure 1: DDVV Approach

3. Model-In-the-Loop tests

The first step of the incremental validation consists into coding in Matlab/Simulink a Functional Engineering Simulator (FES) in order to test the prototype of the GNC ASW designed to achieve the functionality and the performance described in the requirements.

DOI: 10.13009/EUCASS2019-38

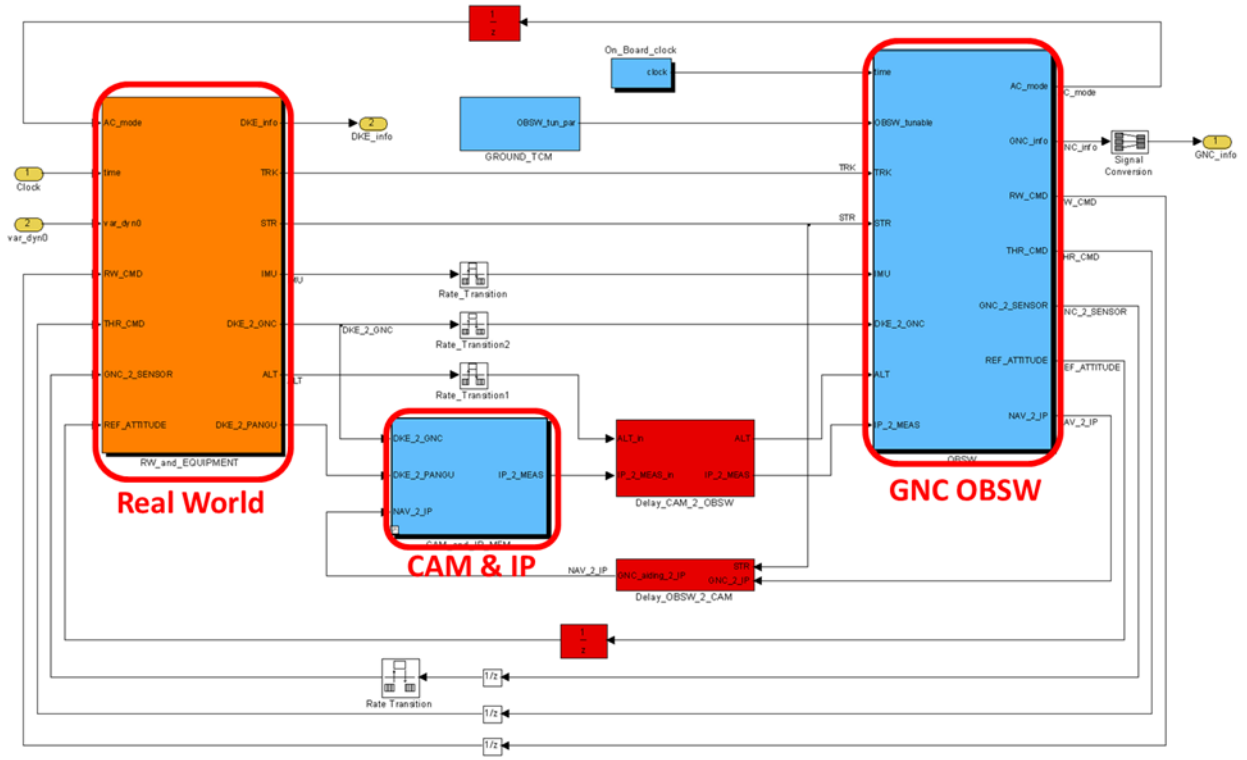

Figure 2: FES and MIL test architecture

Figure 2 shows an example of FES in a MIL test architecture configuration. Up to this stage only a standard PC and Matlab/Simulink code are necessary for testing. The list of the major blocks that can be identified is:

- Real World: Actuators, dynamic and Sensors are simulated in order to close the GNC loop
- GNC OBSW: the GNC ASW + functional/performance model of the OBSW are coded
- CAM & IP: this is a group of functions fundamental for a vision based GNC. In this block the images are generated by tool like PANGU and processed by the Image Processing (not part of the GNC ASW)

The MIL environment is ideal when the code is still unstable and can suffer severe changes/updates. Furthermore, it is the quickest test bench in terms of simulation time, so it is ideal to perform the Monte Carlos Simulations. An intense test campaign can be performed at this level, both to get the required reliability/confidence level and to demonstrate the robustness of the solution varying spacecraft parameters and conditions.

4. Software-In-the-Loop tests

The main objective of the SIL step is to verify the correctness of the SW implementation (obtained through auto-coding technics) with respect to the algorithms implementation of the MIL environment.
The first step for the SIL environment is the generation of the C-code of the GNC OBSW subsystem. The C-code is generated with Simulink Coder / Embedded Coder, which produces readable, compact, and fast C code for use on embedded processors.
The SIL block functionality provided by the Embedded Coder is used; SIL simulation involves compiling and running the same production source code that will be used in the PIL on the host computer to verify the source code in the Matlab/Simulink environment.
The steps followed for the SIL campaign are:

- Generation of GNC-OBSW C-code and SIL block: the process of preparing the MIL models (model configurations, tunable parameters definition, etc.), configuring the Embedded Coder options and generating the C-code and the SIL block is performed in an automatic way by using the ATOCOGEQ Wizard tool. The GNC OBSW subsystem code is generated as a set of C functions that can be easily embedded in both SIL (through the SIL block functionality) and PIL environments (through integration of the generated C-code inside hand-made code for managing tasks and interfaces in the on-board computer). It shall be noticed that the GNC OBSW code used in the SIL is exactly the same as the one used for the next validation step (PIL).

DOI: 10.13009/EUCASS2019-38

An example of the content of the folder generated by the Embedded Coder and containing the C-code is shown in Figure 3.

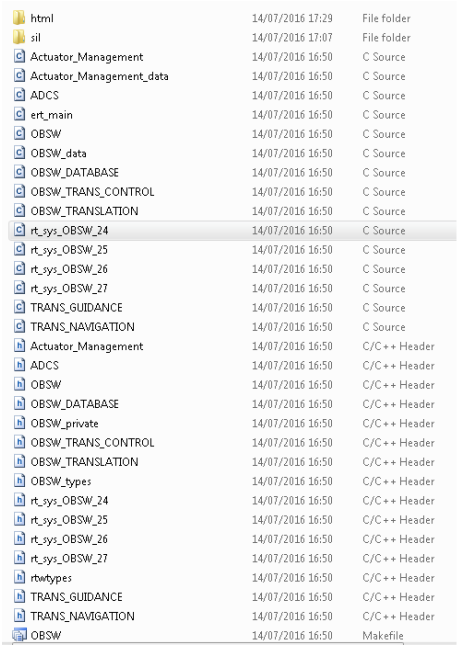

Figure 3: GNC OBSW C-code folder generated automatically

- Integration of the SIL block in the simulator: the integration of the SIL block is easily performed by just inserting it in the complete simulator (including the Real World), see Figure 4.

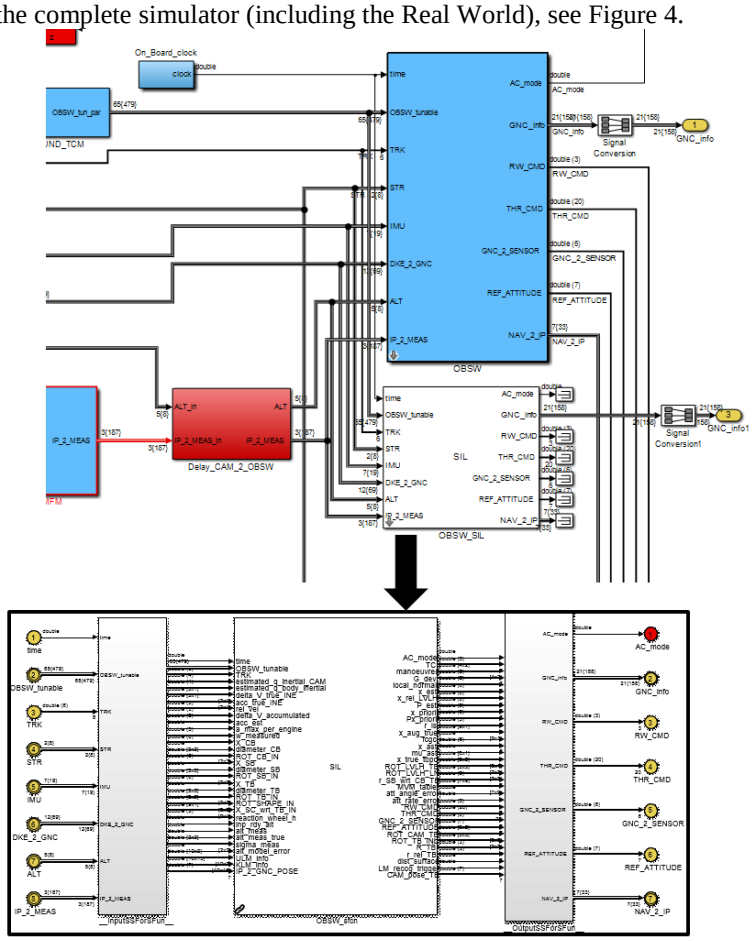

Figure 4: SIL block integrated in the Simulator

DOI: 10.13009/EUCASS2019-38

- Numerical Comparison of the SIL block results: before performing the test runs with the SIL block a direct comparison of the main outputs of the SIL block with the outputs of the original Simulink block is performed. The SIL is run in parallel with the Simulink block (with the same inputs) for a reference case and the outputs are numerically compared in order to detect any problem in the code generation.
- Execution of runs with the SIL block: after the first direct comparison of outputs, the SIL block is run alone in the simulator and the SIL campaign test cases are executed. The results of the runs are analysed by looking at the performances of the system and the behaviour of the trajectories. Additionally a profiling of the execution times is performed in order to identify at early stage the functions which are more time consuming.

5. Processor-In-the-Loop tests

The objective of the PIL test is to demonstrate that the autocoded GNC ASW can run inside a representative space qualified processor (e.g. LEON) and to profile the different functions. The profiling of the execution times allows to identify the most time consuming functions and to correct the implementation if some bottle neck is found (as it may depend on a not optimal implementation from the point of view of the computation time)
The overall development and validation chain based on MIL-SIL-PIL is depicted in the figure below.

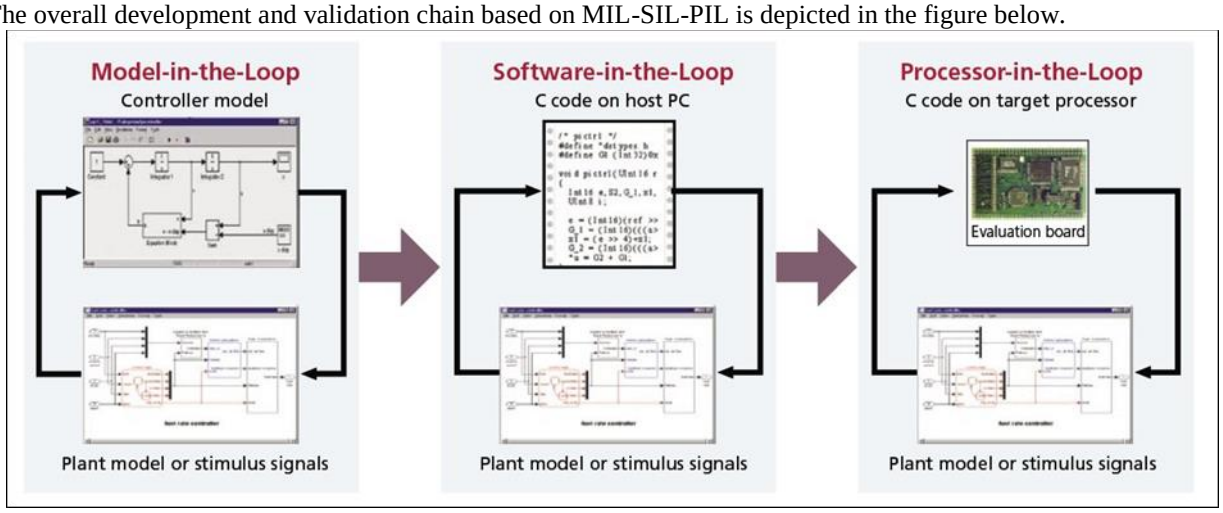

Figure 5: Development and Validation chain: MIL ► SIL ► PIL

It shall be noticed that usually the IP subsystem (if any) is not developed in Matlab/Simulink, but directly in C-code and embedded as an S-function in the MIL-HF environment, therefore for the IP block the SIL version is already present in the MIL simulator.

During PIL tests and thanks to the functions profiling, it is possible to tackle the issue of the Hardware implementation. This is important for the GNC ASW in order to iterate with the OBSW responsible and to be sure to have enough available resources on-board. It is even more important for the IP, because it is usually a heavy algorithm that in several cases might require a separate processing unit (e.g. FPGA with HW acceleration). If the design include an Image Processing Unit, the PIL architecture can already contain it or it can be done at SVF/ATB level.

6. Hardware-In-the-Loop tests

The HIL tests are performed when a representative GNC sensors or actuator is included in the loop. The most typical situation is with a camera in close loop for a vision based GNC. At GMV there are two facilities dedicated to HIL tests:

- HIL-OPT: optical lab that consist in a dark room with a high resolution screen where the synthetically generated images are projected
- HIL-ROB: robotic facility called ***platform-art©*** that thanks to representative illumination conditions and two KUKA robotic arms is capable of recreating space like conditions. In this case there are no synthetic images and the camera take images of mock-ups.

DOI: 10.13009/EUCASS2019-38

6.1 HIL-OPT

The main objective of the HIL-OPT verification campaign is to verify the flight software implementation of the GNC functional modes and the IP on the actual flight processors hardware in a real-time environment with a camera hardware, one of the critical sensor, in the loop.
The HIL-OPT consists in a facility where the camera HW is stimulated through the use of simulated images shown on a screen placed in the FOV of the camera.
In this facility the camera is stimulated with a high resolution screen in which PANGU images are shown. The GMV's optical laboratory (dark room) has been used in order to prevent light disturbances to affect images.
It shall be taken into account that for a correct stimulation of the camera the monitor shall have a bigger resolution in order to avoid aliasing effects.
A major parameter of the display device is the image resolution, i.e., the size of the displayed image in pixels. In terms of resolution, the higher the resolution the better. By experience, 2×2 display pixels can stimulate 1 detector pixel (see Figure 6) in a satisfactory way. The 4:1 ratio is appropriate for HILOPT testing as

- It is sufficient to avoid the noticeable aliasing that may show up when the ratio of display pixels to detector pixels is near 1:1.
- It is an acceptable trade-off between approximating the continuity found in natural images and stimulating a large part of the detector.
- It smooths the stimulus granularity also in brightness.

The selected display device is the Samsung U24E850R monitor, a LED-PLS screen of 24'' diagonal and 3840X2160-pixel resolution (UHD).

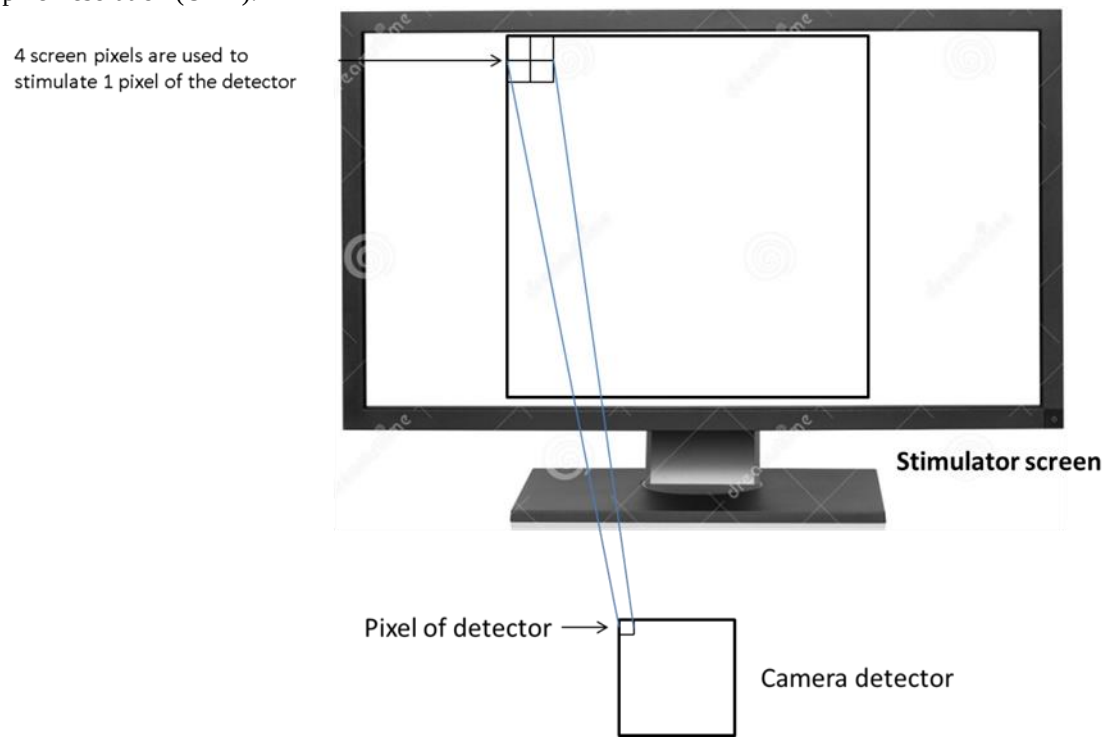

Figure 6: HILOPT camera stimulation concept

An example of the architecture of the HIL-OPT testbed is depicted in Figure 7. It is based on the re-use of all the components of the PIL test-bench with the addition of the HW camera and the scene generator facility provided by the GMV's optical laboratory.
From the point of view of the GNC and IP system, the items under test, the transition from PIL to HILOPT is transparent as all the interfaces with these systems are exactly the same, and the only difference between PIL and HIL-OPT is that the images provided to the IP come from an HW camera and not from the image generator SW directly (as in PIL).

DOI: 10.13009/EUCASS2019-38

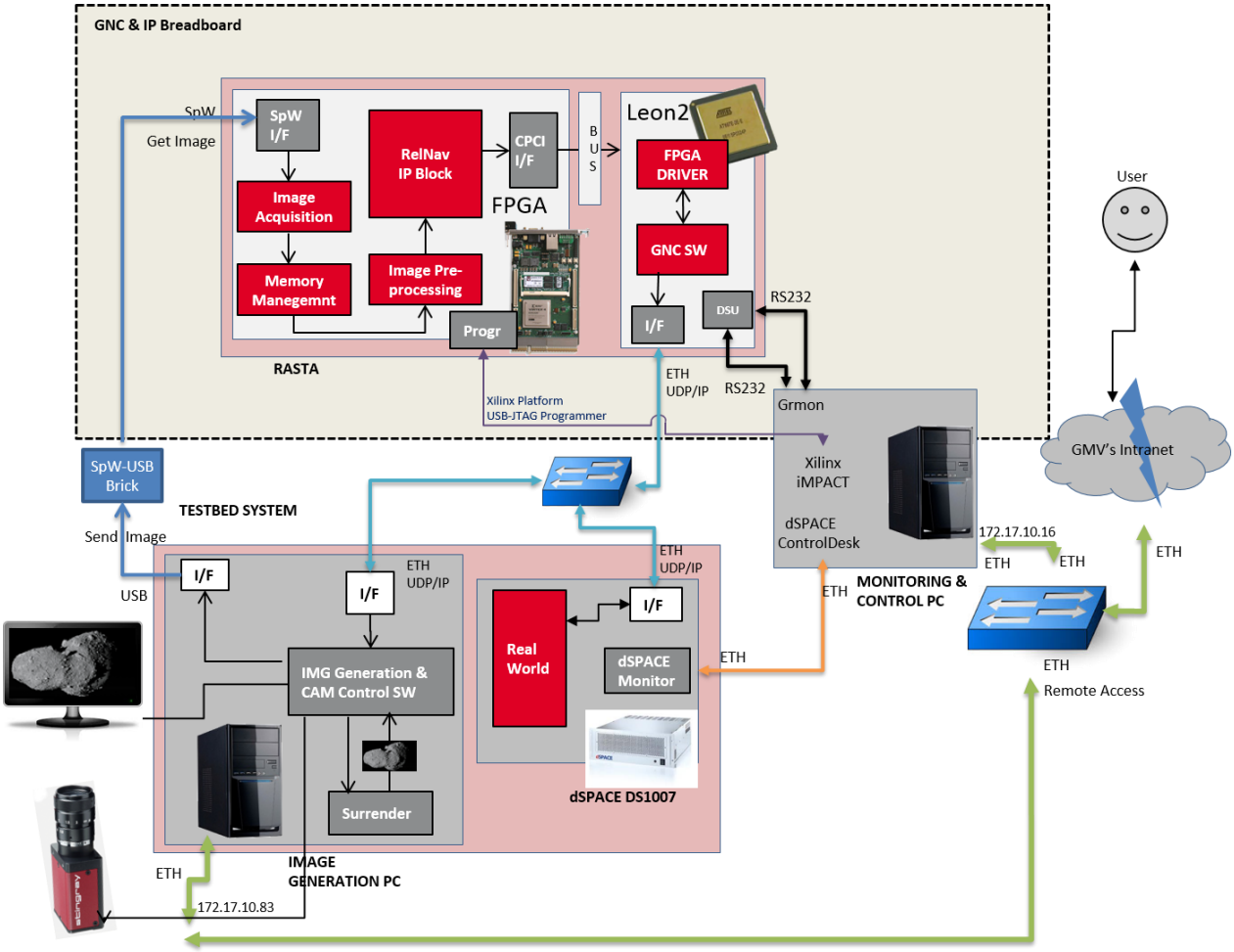

Figure 7: HIL-OPT Test-bench Architecture

6.2 HIL-ROB

The main objective of the HIL-ROB verification campaign is to verify the flight software implementation of the GNC functional modes and the IP on the actual flight processors hardware in a real-time environment with representative hardware (e.g. camera, laser altimeter) in the loop.
The HIL-ROB is a facility where the camera HW is stimulated through the use of a mock-up placed in the FOV of the camera and sensed by the laser altimeter.
The architecture of the HIL-OPT testbed is depicted in Figure 8. It is based on the re-use of the components of the HILOPT test-bench, except for the image generator and the stimulation screen, with the addition of a laser altimeter if needed, the robotic arms and the mock-ups; it is realised adapting the GMV's ***platform-art***© facility to the scenario.
From the point of view of the GNC and IP system, the items under test, the transition from HILOPT to HILROB is transparent as all the interfaces with these systems are exactly the same, and the only differences between HIL-OPT and HIL-ROB are:

- The camera is stimulated by a scene of the mock-up
- The altimeter measurements come from a real hardware sensor, instead from the simulated sensor (if the altimeter is included in the sensor suite).
- The Real World computer sends commands to the ***platform-art***© "Motion Control System" for the reproduction of the camera trajectory and illumination source movement.

DOI: 10.13009/EUCASS2019-38

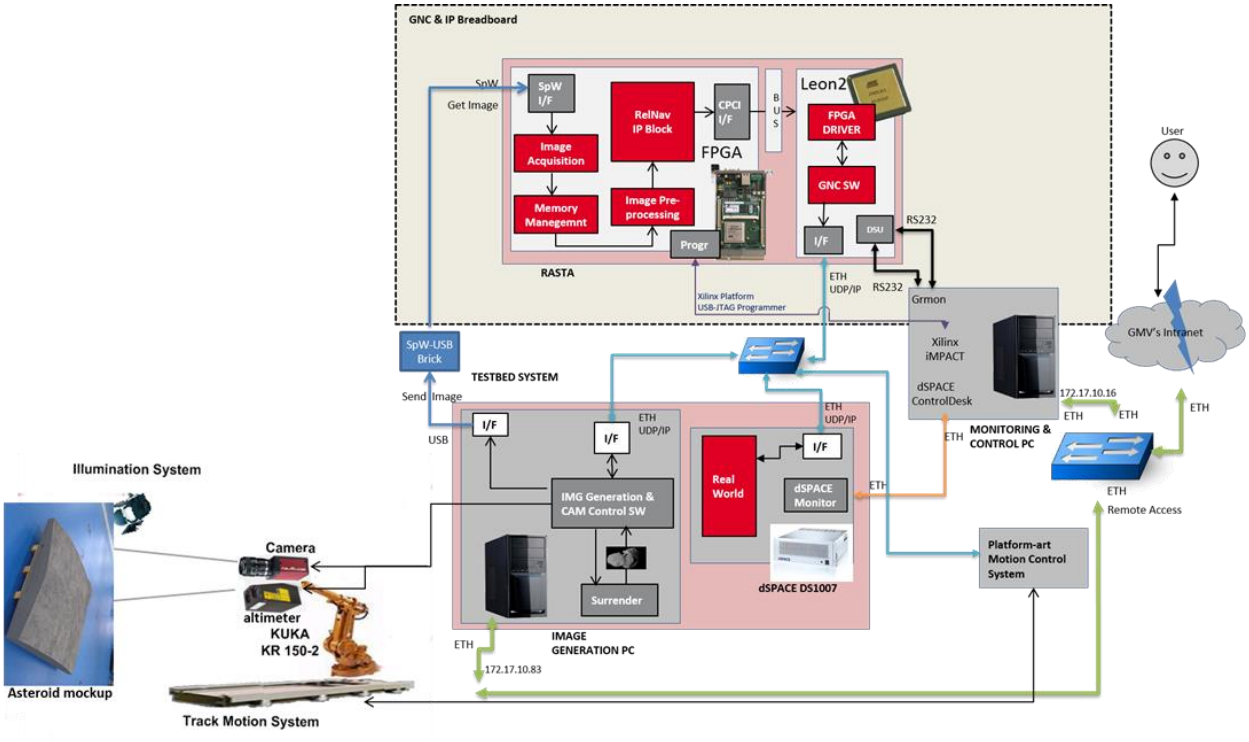

Figure 8: HIL-ROB Test-bench Architecture

The figure below shows an example of the ***platform-art*©** set-up during the HIL test of the AIM phase B1 project.

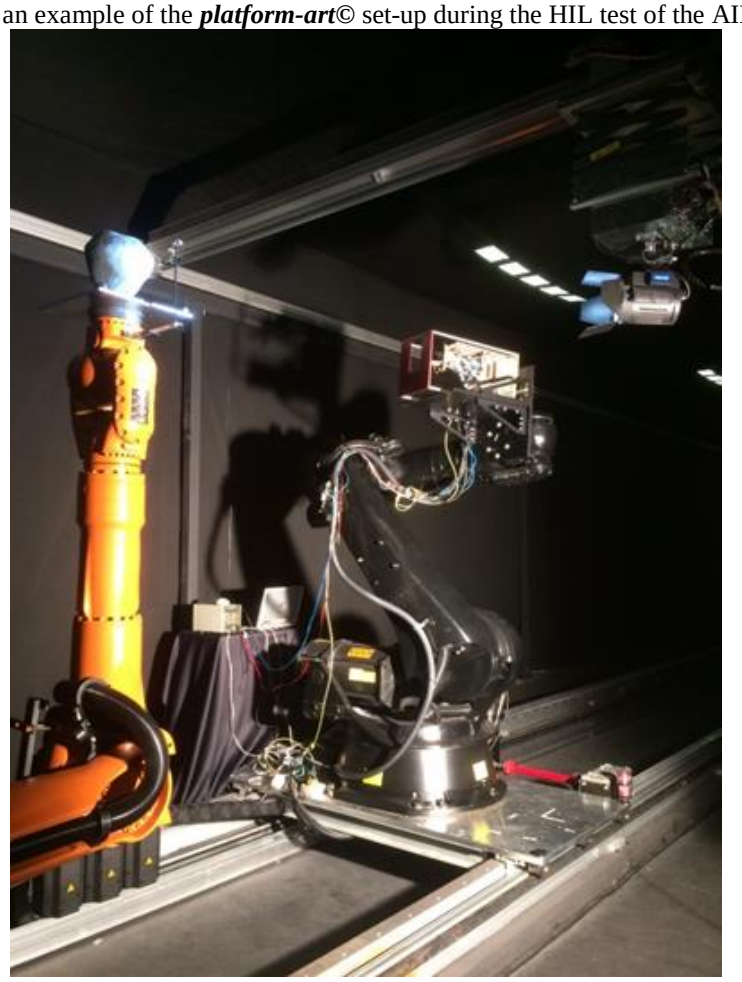

Figure 9: ***platform-art*©** setup for AIM HIL tests

DOI: 10.13009/EUCASS2019-38

7. Conclusion

The GMV approach for the incremental DDVV dedicated to a GNC subsystem has been described, including practical examples. It is important to remark some general rules that shall be followed in order to optimize the process:

- Use the MIL architecture in order to perform the most intensive test campaign (e.g. Monte Carlo).
- Identify some reference tests that shall be repeated at SIL, PIL and HIL level. The purpose is to validate the code already presented at MIL level and, as a consequence, all the results of the MIL test campaign
- Perform regression tests every time the code required a modification in order to proceed with the incremental validation

GMV has been following this approach for more than 10 years, raising TRL of autonomous technologies up to 5/6. Apart from ground based validation, for this kind of technologies, it is good practice to perform on-flight validation of the autonomous technique. The safest approach is to use a function/mode off-line, collect data, cross-check on ground and if the performance is satisfying to give on-board authority. This validation approach can even drive some GNC functions design. For example, some functions can be implemented as semi-autonomous, with specific control/monitoring, thresholds so to be fail-safe during phase with on-board authority.