SCOE for IXV and ExoMars GNC

Enrique Rodríguez García [(1)], *Antonio Ayuso Barea* [(1)], *Ignacio Barrios Tascón* [(1)], *Ignacio de Miguel Matiacci* [(1)], *José María de las Casas Gilarranz* [(1)], *Pablo Giménez González* [(1)], *Vicente Fernández* [(2)] *Pedro Palomo* [(2)], *Rodrigo Haya Ramos* [(2)], *Cristina Parigini* [(2)]

(1) *SENER Ingeniería y Sistemas, S.A., Severo Ochoa, 4, 28760 Tres Cantos, Madrid (Spain), Email:*
enrique.rodriguez@sener.es
(2) *DEIMOS SPACE S.L.U., Ronda de Poniente, 19, 28760 Tres Cantos, Madrid – (Spain), Email:*
vicente.fernandez@deimos-space.com

INTRODUCTION

Verification is a fundamental stage in the design process of any space programme in general, and any Guidance Navigation and Control subsystem (GNC), in particular. The GNC-SCOE contains all the hardware and software elements needed to test a GNC. It is used to virtually close the satellite attitude and orbit control loop on earth. The GNC-SCOE will be used for the design of the GNC, and also during its Assembly Integration and Verification (AIV) campaign.

SENER has developed a generic modular architecture for a GNC-SCOE. The GNC-SCOE is based on the simulation models of all GNC sensors and actuators, and has also a model for Dynamics, Kinematics and Environment (DKE). This modular architecture allows replacing simulation models by real hardware models. This paper presents the application of this generic architecture to the GNC of the Intermediate Experimental Vehicle (IXV) and to the GNC of the ExoMars Entry-Descent and Landing Demonstrator Module (EDM).

The hardware architecture is based on modular PXI boards that provide the SCOE with input/output ports as well as timing capability. The software architecture is based on two main elements: the SCOE Controller SW and the Real Time Simulator (RTS).

The GNC-SCOE can also be integrated with the rest of the Electrical Ground Support Equipment (EGSE) for system verification and for Telemetry/Telecommand (TM/TC) handling, and is supporting the commonly used standards protocols an languages as PUS standard, EDEN protocol and TOPE language.

GENERAL DESIGN

The GNC-SCOE is therefore simulating all the environment and sensors used for the GNC verification. Starting at the early development phases of the GNC and during the whole development it can be used in three main stages:

- The pure GNC software verification.
- The verification of the GNC in real time with the On-Board Computer in the loop.
- The verification of the GNC having real hardware units in the loop.

The GNC SCOE uses hardware and software for the different development stages to do the activities listed below:

- Measuring satellite Hardware in the Loop (HIL) actuators activities.
- Acquiring commands for satellite simulated actuators.
- Calculating the corresponding effects on the satellite attitude and position.
- Emulating outputs from simulated sensors.
- Stimulating HIL sensors interfaces according to the calculated attitude and position status.

Apart from connecting all relevant GNC interfaces, the GNC-SCOE can be integrated with other external EGSE elements and a Central Check-out System (CCS) emulator for remote control. Finally it provides a local control through a specific Man-Machine Interface (MMI).

Hardware Architecture

The GNC-SCOE is based to the maximum possible extent on commercial “off-the-shelf” (COTS) equipment minimizing the custom-made hardware development wherever possible.

The hardware architecture design is compliant with:

- Spacecraft Grounding Philosophy. The GNC-SCOE follows a grounding approach compatible with the spacecraft grounding to avoid undesired wrong measurements, and safety problems.
- Spacecraft Interface Protections. The GNC-SCOE is interfacing the GNC sensors, actuators, and on board computer. Special care must be taken to avoid any damage to the flight units, but also to protect the GNC-SCOE interfaces. A dedicated Built In Test (BIT) is generated for this purpose.
- Environmental operation restrictions. According to the mission requirements, the GNC-SCOE can be built to be EMC compatible, to follow the CE rules, to be operated in a clean-room up to ISO-7, and to have the harness entering a bio-burden controlled area.

The main hardware elements are presented hereafter:

SCOE Rack

The main hardware element is the SCOE Rack, which will be placed close to the GNC under test.

PXI Rack

This is a 3U high and 19” wide PXI Rack. It is composed by a PXI Chassis, a PXI Controller, a PXI Synchronization board and other instrumentation boards inserted in the available slots.

The PXI Chassis provides the mechanical support of the PXI rack. It also contains the power source, that supplies all the PXI systems, and the PXI backplane that offers advanced timing and synchronization features.

The PXI Controller is an embedded computer running a real time operating system. This is the engine of the GNC-SCOE, since it is where the software runs.

The PXI Synchronization board enables to synchronize PXI systems. It is used to synchronize the GNC-SCOE with the spacecraft, with other EGSE and with the external CCS.

The instrumentation COTS boards, required for all the on board equipment interfaces, provide and acquire the interface analogue and digital signals, as well as the communication buses.

SCOE MMI Host

It is the computer that hosts the SCOE MMI software to provide command, monitoring and archiving tools for the GNC-SCOE. Normally the SCOE MMI Host is a single computer where all the software is installed, but it can also be enlarged with additional computers or archiving media in case of additional needs for data archiving or for more developers working in parallel.

SCOE CCS Host

It is the computer that hosts the CCS software and the test language where the AIV engineers will develop the test scripts.

SCOE Harness

The SCOE Harness has the required length to reach the spacecraft connectors, providing all required signals to the On Board Computer, and also to the units, in either simulated or HIL configuration. It can also provide test points, where additional equipment could be attached to monitor signals without affecting their values.

SCOE Local Area Network

The GNC-SCOE LAN connects all relevant GNC-SCOE hosts. The SCOE LAN provides enough capabilities so that the SCOE MMI can display and archive simulation data during run time. The usage of an independent LAN isolates the GNC-SCOE from external network traffic and makes possible to have a specific access control policy.

Software Architecture

The main software elements of the GNC-SCOE are described below. The main software elements run in different computers, but the core software element, composed by the Simulation Software and the Communication Software. They both run on the SCOE Controller, which is located in the SCOE-Rack.

Monitoring and Control Tools

These tools include all the SCOE functions for controlling, commanding and monitoring. These functions can be executed locally, from the MMI Host located in the SCOE rack, and remotely, from the CCS Host. The following functions are provided:

- MMI displays and controls.
- Built in test.
- Logging and archiving software.

Simulation Software

It is the engine of the GNC-SCOE, which provides all the needed test capabilities, to allow verification at unit level and subsystem level. The simulator software is structured in models. It contains sensor models, actuator models, and DKE models. Models can implement a complete simulation for a certain unit, or just the needed signals to support the unit hardware in the loop. It is able to run in real time, and provides visualization and logging of all relevant data. These models can be generated in Matlab/Simulink, C language or Labview.

Real Time Software

It is the software required to integrate the simulation software with the real time operative system (RTOS). The RTOS is called LabView Real-Time. This software is a PXI controller real time compatible OS from National Instruments, and is deployed along with the SCOE in the PXI Rack.

The elements of Real Time Software are:

- LabView environment: the SCOE is developed in LabView using Real Time module and deployed in the PXI rack. The LabView environment is the responsible of controlling the DLL iterating function.
- Generated DLL (part of Real Time Software): a DLL calling the functions of all the models is going to be generated. The entry point of this DLL is a single function that receives the iteration number and flags (to know what models are currently active), and know what functions are to be called depending on those parameters. This schedule respects the real time constraints.
- Simulation Models: the models called by the DLL to execute the functions during the Cyclic Executive. In order to have an executable software independent from LabView, the models can be compiled generating a DLL that will be called from the LabView environment. Updated versions of models can replace older versions. The procedure consists on using any suitable compiler, such as Visual Studio or GCC capable to generate windows DLLs. The only restriction is to keep the same interface of the model. To maintain the original interfaces, the models are designed keeping some spare variables to be substituted in the future by new entry or exit parameters.
- Data pool: it is a shared memory on the PXI Controller, containing the inputs and outpus of the models, thus allowing the exchange of data between models.

The normal cycle that explains the process of one cycle of simulation can be divided into the following steps:

- In a timed loop with high fidelity, the DLL cyclic function is called.
- The parameters for this function are the number of iteration, flags to select/deselect the models and data structures to feed the sensors.
- The DLL decides, based on the iteration, which models have to be executed.
- Once executed, the data pool is updated.
- GNC- SCOE uses the Host Layer to read the data and feed again the call of DLL for the next iteration.

Communication Software

It provides the communication layer for the Simulation Software, establishing communications both with the hardware connected to the GNC under test, and with the external control and display systems (MMI and EGSE).

- Serves as EGSE router, for remote control.
- Handles the communications with the MMI for local control.

- Includes all related software interface for analogue and digital signals, and handles MIL-1553 bus traffic, as well as the synchronization tasks.

MMI Software

This software is the interface for the user. It provides the following functionalities:

- Local control of the SCOE.
- Simulation control.
- Log control.
- Error injection control.

CCS Software

This is the software based on SCOS-2000, used for operations by ESA, here used for AIV campaigns. The GNC SCOE can include a SCOS-EGSE as provided by ESA.

Simulation Software Development Procedure

To help the development of the Simulation Software used in the GNC-SCOE, a Software Development Environment is used. It is based on a set of common commercial software tools like compilers, converters, graphic languages, download tools, etc. Several different programming and developing common languages are supported, like Simulink and C. The simulator modules are merged into a single Simulation Software, and integrated together with the Communication Software to get finally a single executable file, that is executed in real time on the SCOE Controller.

APPLICATION TO IXV

A GNC-SCOE has been designed to test, verify and validate the entire GNC subsystem of the Intermediate Experimental Vehicle (IXV) in real time.

The GNC SCOE is developed in two versions, both executing GNC software in real time. The first version is intended to perform preliminary verifications on the GNC software by using an advanced delivery version of the On Board Software (OBSW), running in an On Board Computer simulator, based in TSIM to allow the GNC execution. The second version provides a real time interface with the Avionics RIG and other EGSEs, to interface the IXV hardware both at the Avionics/GNC Test bench and at the IXV PFM.

In the first version, the GNC units models integrated in the RTS are fully simulated models for the sensors (IMU and GPS) and actuators (FPCS and RCS).

As the RTS is based in a modular architecture, it is possible to select different configurations by replacing models by others with hardware in the loop interface capabilities. This is the way that the second version of the GNC SCOE is used. It still provides the same simulated models, but they can be replaced by specific ones to allow the GNC SCOE to interface the flight hardware (OBC and POW) as well as the other EGSEs used to stimulate hardware in the loop units (IMU-Lab and GPS Spirent) or acquire signals from this hardware.

The IMU can be fully simulated by a software model or introduced in the loop by interfacing the IMU-Lab via Reflective Memory. The GPS can be also simulated or stimulated by using the GPS Spirent. The RCS pulses are always acquired by using FCV dummies. Finally the body flap chain simulation model can be used for the FPCS, or a Flap Model can be chosen to allow the EMA/EMACU information (commands and position values) acquisition.

Fig. 1 shows the architecture of this design:

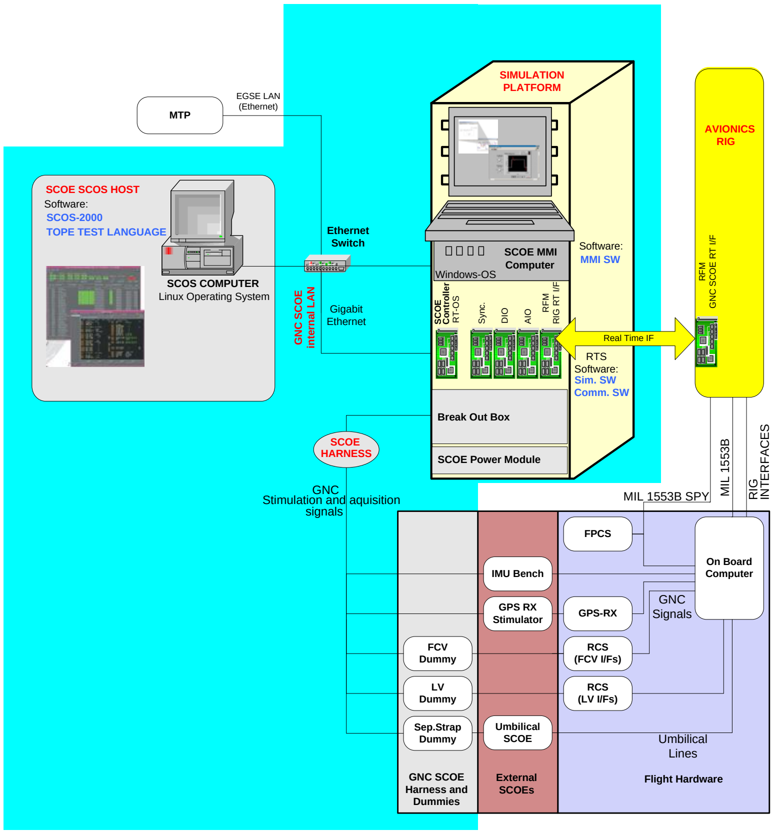

Fig. 1 IXV GNC-SCOE Architecture

APPLICATION TO EXOMARS

A GNC-SCOE has been designed to test, verify and validate the entire GNC subsystem of the ExoMars Entry Descent and Landing Demonstrator Module (EDM) in real time.

The GNC-SCOE provides a Front End Electronics (FEE) to interface the EDM hardware (CTPU and RTPU) as well as other EGSEs used to stimulate hardware in the loop units. The GNC SCOE executes a Real Time Simulator (RTS) which features a model of environment and dynamics (ENVDYN) and integrates models for GNC units. The RTS provides simulation capabilities for IMU and SDS sensors; acquisition of FCV pulses from the EDM RTPU, and the needed stimuli information for the RDA EGSE, which is interfaced by means of a reflective memory to stimulate the RDA.

The GNC SCOE will allow local operation, from a Man Machine Interface (MMI) and remote operation from a Central Checkout System (CCS) through Ethernet LAN. The RTS is also ready to acquire specific events through interfaces with external EGSEs (PWR SCOE), in order to change the behaviour of the ENVDYN.

To complete the data provided to the AIV engineer, the GNC SCOE provides Monitoring and Control Tools to record the traffic of digital busses, display, log and archive test data. It is also able to perform Error Injection in the RTS variables to allow simulation of errors.

From the hardware point of view, the GNC SCOE architecture is based on COTS products. Flight hardware will be protected by dedicated electronics.

From the software point of view, the GNC SCOE SW follows a modular approach allowing the integration of specific ExoMars EDM Real Time Simulator modules with the rest of SCOE software.

Fig. 2 presents the architecture of this design:

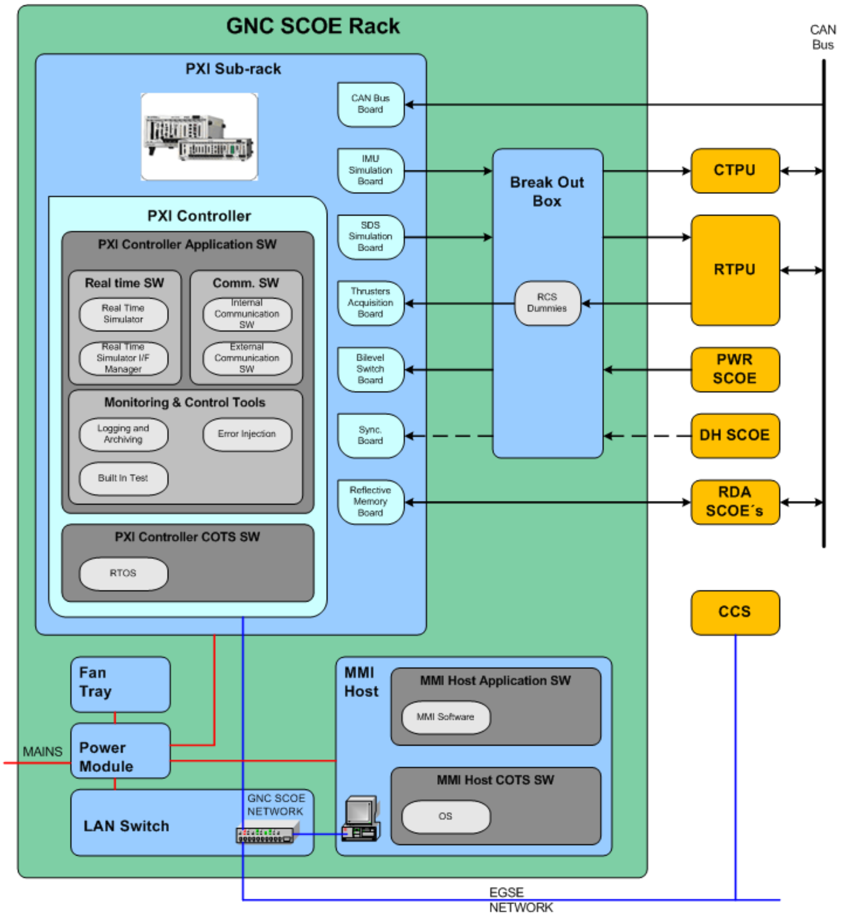

Fig. 2 ExoMars GNC-SCOE General Architecture

FUTURE DEVELOPMENTS

The possibility of extending this modular architecture to other EGSE by SENER, such as Data Handling SCOE and Power SCOE, is being currently analyzed.

CONCLUSIONS

SENER has proved to have a flexible GNC-SCOE architecture design able to adapt to different types of missions. This architecture has been used for IXV and ExoMars missions and is actually being offered for other projects.

In the future, this architecture could be adapted to other EGSE, such as Data Handling SCOE and Power SCOE.

REFERENCES

[1] C. Vicente, I.Barrios, B.Lyautey, D. Gherardi and E. Rodriguez, “Integrated Test Bench for rapid GNC design, verification and validation”, 4th ICATT, May 2010.

[2] E. Rodríguez, A. Ayuso, C. Vicente, D. Gherardi, R. Sánchez and I. Barrios, “Integrated Test Bench for rapid GNC design, verification and validation”, SESP, September 2010.

[3] W. Fehse, “Automated Rendezvous and Docking of Spacecraft”, 1st ed. Cambridge University Press, pp. 362-424, 2003.

[4] J. Eickhoff, “Simulating Spacecraft Systems”, 1st ed. Springer Series in Aerosp ace Technology, 2009.