SESP 2010

Session:	Session: Functional Engineering Simulators (04)
Type:	Concurrent Session
Date:	Tuesday, September 28, 2010
Time:	11:30 - 13:00
Room:	Newton
Chair:	
Co-chair:	
Remarks:	

Seq	Time	Title	Abs No

1 11:30 The IXV GNC Functional Engineering Simulator

Fernandez, V.; De Zaiacomo, G.; Mafficini, A.; Peñin, L. F.
DEIMOS Space S.L.U., SPAIN

The Intermediate Experimental Vehicle (IXV) programme is the next milestone in the history of europen re-entry demonstrators. The programme is now in the C2 phase, when many subsystems are undergoing their final detailed designs. Among them, the Guidance Navigation and Control (GNC) is a key subsystem of the vehicle that must pass through exhaustive verification process to ensure the success of the mission. Our paper will describe the IXV GNC Funcional Engineering Simulator, which has become the reference simulator for all re-entry GNC design and validation activities within the programme. It is a clear example of a simulator to support the verification of critical elements of a baseline design that is later upgraded with phase C/D models and data.
The IXV-FES is a true 6 DOF re-entry simulator. In fact, it allows the GNC designer to carry out 3 DOF (assuming ideal control) and 6 DOF (with the actual control) analyses. The simulator supports the exoatmospheric phase of the mission, from vehicle separation to the start of the guided trajectory, and the re-entry phase, from the entry interface point to parachute openning, which occurs at about 1.6 Mach. It is also possible to run end-to-end simulations, comprising both phases. The IXV-FES, which was born to support the GNC design verification activites of the B2/C1 phase, is now evolving to sustain the requirements of the current phase, integrating more detailed environmental models to validate the final GNC design that will actually control de re-entry of the vehicle. The IXV-FES was developed and validated by DEIMOS Space, following a development specification provided by Astrium Space Transportation. The development of such simulator under the constraints of the IXV programme schedule in a cost-effective way wouldn't be possible without the reuse of simulation models and infrastructure. In this respect the SIMPLAT infrastructure has played a relevant role. SIMPLAT is a simulation infrastructure designed and developed by DEIMOS Space for the production of functional engineering simulators. The SIMPLAT infrastructure is based on the MATLAB™/Simulink™ modelling & simulation environment and provides all the basic functionalities needed by a FES tool, so that project-specific elements can be rapidly built on top of it. SIMPLAT operation largely relies on its XML database, which stores model, scenario and simulation parameters. SIMPLAT includes Monte Carlo simulation and analysis capabilites and several generic toolboxes and blocksets. The simulator includes generic post-processing functions which proved very useful to compute the required GNC performance indices.
The SIMPLAT infrastructure supports the auto-coding of Simulink models for the dSPACE™ and Rapid Simulation targets. Thus, the IXV-

FES will easily evolve into a real-time test bench (RTTB) for assessing the real-time performance of the GNC algorithms.
Finally, we will address specific problems found during the actual execution of IXV GNC validation plans and the solutions implemented in the IXV-FES simulator to cope with them. Re-entry GNC simulation is highly demanding in terms of hardware resources. The simulator includes high fidelity environmental models, full 6-DOF vehicle dynamics and complex guidance, navigation and control algorithms which impose a substantial computational load to the system. In terms of data storage needs, Monte Carlo simulations often comprise 500 repetitions at least, using a short simulation step of a few milliseconds along more than 1000 seconds and can generate up to 100 Gbytes of data for post-simulation analysis.

2 12:00 An Integrated Framework for Reconfigurable Mission Performance and Functional Engineering Simulation Tools

Araujo, J.; Hormigo, T.; Camara, F.
Spin.Works Lda., PORTUGAL

Spin.Works has developed a framework for preliminary mission performance simulators and for functional engineering simulation tools of varying degrees of freedom and fidelity, with the primary purpose of serving as a development and test platform of GNC algorithms especially in the early stages of the development life-cycle for space systems. A generic decomposition and interconnection of the various physical models required to simulate a dynamic system (Dynamics and Kinematics, Environment, GNC, Aerodynamics, etc.) has been defined, in order to guarantee maximum reusability, support the continuous development and integration of new models in the existing framework, minimise the investment made, as well as expedite the tailoring process involved in implementing simulation tools for specific applications. Within this simulation framework, the aggregation and substitution of different model combinations can be made with minimal programming effort, and switching between multiple combinations of models (e.g., gravity-atmosphere-vehicle-actuator-GNC models) can be made prior to each simulation during a configuration step. A physical model database is set up separately, which includes all usable combinations of models, as well as a larger set of individual models. This allows incrementally testing a GNC system with progressively more complex/complete models using the same underlying data flow architecture.

This simulation framework has been successfully implemented in the scope of ESA projects related to Entry, Descent and Landing, and applied to different vehicle configurations (capsules, lifting bodies, space planes, powered landers), for various mission phases (namely Entry, Terminal Area Energy Management and Landing), and operating in conjunction with external tools (high-fidelity atmospheric models, artificial scene generation tools and visualization models, complex sensor models, software running in external hardware, etc.). This concept has also been applied to non-space activities, namely for an unmanned aerial vehicle developed by Spin.Works which is currently undergoing flight-testing.

3 12:30 The Avionics System Test Bench, Functional Engineering Simulator: New Developments in Support of Mission and System Verification

Rebelo, J.[1]; Wijnands, Q.[1]; Blommestijn, R.[1]; Pace, F.[2]
[1]ESA/ESTEC, NETHERLANDS;
[2]GMV, SPAIN

The Avionics Systems Test Bench (ATB) is an ESA/ESTEC internal development of an avionics system test bench aiming to support the verification and validation of space avionics issues in a representative context. This system is an evolution of the Virtual Spacecraft Reference Facility (VSRF), which main focus was a TSIM-based Software Validation Facility (SVF). This led to a bottom-up approach for the design of the Functional Engineering Simulator (FES), heavily relying on heritage of simulation models and modules, which resulted in a configuration highly tailored to the VSRF Reference Mission.

In the context of supporting missions like the European Student Moon Orbiter (ESMO), Galileo and SGEO, functionality-wise and quality-wise improvements were needed. This paper describes the developments done on the FES configuration with the aim of verifying AOCS/GNC algorithms, subsystem trade-off analysis, real system "margins" and FDIR algorithms.

Taking into account the modularity requirements given in ETM 10-21 and referring to the Functional Engineering Simulator Enhancement activity as carried out by GMV a generic and modular simulator infrastructure based on Matlab/Simulink® was created. The intermediate results of the ongoing "Space Simulation Reference Architecture" were used as guidelines for the new implementation to ensure maximum consistency.

A new set of parameterized dynamics, kinematics, environmental, sensors and actuators models was developed. An analysis was made to compare the use of Simulink library block and reference model, highlighting the advantages and disadvantages of each approach. Given the level of modularity and the architecture of the proposed simulator the new subsystem are implemented using Simulink library blocks. The models' interfaces are strictly defined using Simulink Bus objects. This approach provides a check both on the signal names and dimensions for the blocks' inputs and outputs. Each model's help documentation is accessible during design-time and a direct link between the software requirements and the model is also implemented, allowing the access to the model from the requirements document and vice-versa.

To ease the simulator instantiation and the design of new models all the parameters, interface definitions and data dictionary are kept in a database to keep a consistent and traceable baseline for each simulation campaign. The data dictionary adds logging control capabilities and the possibility of including metadata, such as units and reference frame, to the signals in the simulator.

For the testing of FDIR algorithms and failure situation analysis a failure injection mechanism is implemented. Each spacecraft subsystem model has a failure condition included, which tries to replicate the possible hardware failures. These failures are enabled by using a Matlab script which triggers the fault, either on time or on specific events in the simulator. The scenario is integrated in the top-level simulator architecture which permits the testing of nominal cases and different failure situations without making any changes to the models or parameters.

All the models were developed following a Matlab coding standard and verified using the Mathworks Model Advisor tool to ensure full compliance. This verification is integrated in a complete unit test framework which also provides information on the performance, behavior and test coverage for each model.

The publication and sharing of results is also standardized by including a plotting and auto-reporting facility. The plotting facility provides functions to create the most common plots in a spacecraft simulator and provides an open framework to be extended to create new plots, depending on users needs.

A TCP/IP connection with the OpenIGS visualization system is also implemented to integrate the simulation and visualization environments during run-time. This way, the simulation results can also be analyzed by observing the 3D visualization system.

The paper describes in more detail the above mentioned enhancements, the trade-offs and the results of the enhancement activity. Also the easy instantiation of a FES for new missions, being either geostationary, navigation or interplanetary missions, and new mission scenarios is highlighted.