End-to End simulation environment: from GOCE E2E simulator to E2E Simulators for Exomars and NGGM

De Sanctis, S[(1)], Parisch, M.[(1)]

[(1)]Thales Alenia Space Italia, ITALY
Strada Antica di Collegno 253, 10146 Torino, ITALY
Email: simona.desanctis@thalesaleniaspace.com
Email: manlio.parisch@thalesaleniaspace.com

INTRODUCTION

The TAS-I experience confirmed that an End-to-End SW (E2ES) simulator is a fundamental design analysis and verification tool for missions providing a high-fidelity performance prediction of the overall system. An E2ES is foreseen to be set up since the early stages and continuously updated according to the project evolution, including the characteristic of the payload and spacecraft equipment, derived from tests (when possible) or detailed analyses.

Throughout a project life cycle the E2ES is progressively updated and utilized for the following functions:

- Mission performance assessment and error budget computation.
- Simulation of equipment behaviour under different environmental conditions especially in case it can not be testable on ground
- Validation of the requirement apportionment to the error contributors.
- Validation of the control design.
- Design and validation of the in-flight calibration methods.
- Assessment of the spacecraft failures impact on the mission performance.
- Simulation of the spacecraft dynamics and equipments for hardware-in-the-loop tests.

The concept has been firstly applied to GOCE programme. It has been then upgraded and it is now the base of the new performance simulators developed in TAS-I for supporting the current studies relevant to the Formation Flying simulation (NGGM: Next Generation Gravity Missions) and the Entry Descent and Landing (EDL) simulation (Exomars).

This paper describes the evolution from the GOCE E2ES structure up to NUMES, the current TAS-I numerical simulation frame concept aimed at supporting the full life of a project.

HINTS ON THE GOCE EXPERIENCE

The E2ES of GOCE is the tool utilized for producing the most reliable prediction of the GOCE in-flight performances. It includes a high-fidelity model of the gradiometer, of the GPS, of the spacecraft dynamics and of the orbital environment (gravity field, residual atmosphere).

In synthesis the simulator must be able to generate the overall data output produced during the GOCE science mission which is a 60 days measurement campaign consisting of about 22 GB data for the gradiometer, star tracker, GPS receiver telemetry and for the ancillary data and 45 GB for the simulation post-processing data consisting of the elaboration of the raw (Level 0) measurements and generation of the Level 1b products (gravity gradients, orbit), including the Gradiometer calibration parameters.

The very stringent requirements on number crunching performances and reliability drove the selection of the HW & SW environment:

- The HW platform utilized in TAS-I is a workstation with Dual Xeon Dual Core 3.2 GHz, 8 GB RAM, GNU/Linux OS RHEL 5.5 (kernel 2.6.18);
- The E2ES is written in C and Fortran high level languages part of which is auto-coded employing Mathworks Simulink and Real Time Workshop exported C code;
- The 60 days data simulation post-processing relies on dedicated C code stand-alone programs and Mathworks Matlab scripts.

Fig. 1 sketches the GOCE E2ES SW components together their real HW counterparts:

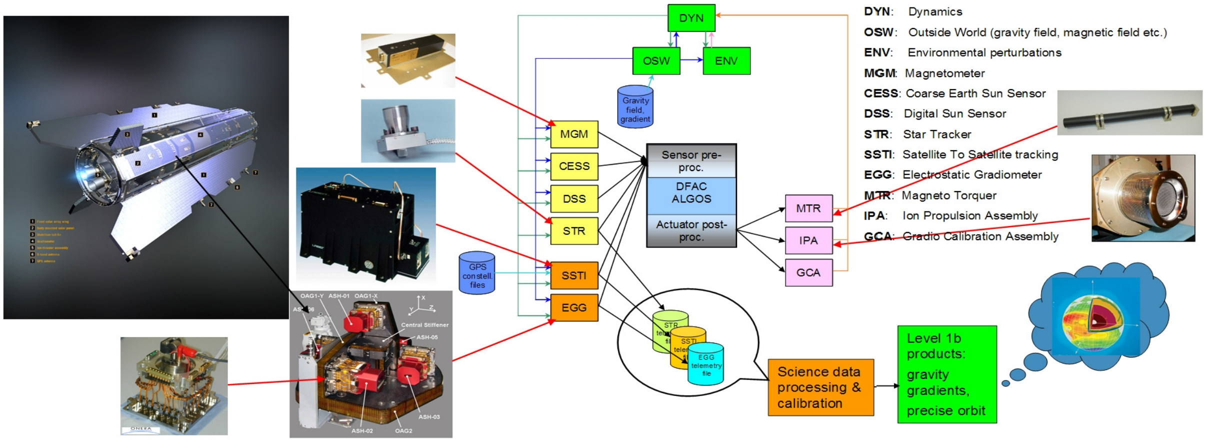


Fig. 1 GOCE E2E Simulator Scheme

The simulation runs 6.5 times faster than the real time then 60 days require about 10 days of waiting time. The E2E Simulator can be compiled and run under the following operating systems: GNU/Linux, Unix, SGI-IRIX, Windows/Cygwin, Mac OS X.

These implementation requirements are applicable for the GOCE E2ES:

- Modularity for:
  - easy replacement of models with different accuracy levels as they are made available along the project life;
  - easy integration of SW written by different teams and produced by autocode generators (e.g. RealTime Workshop);
  - the distribution of computational burden in parallel processors/processes/threads;
  - Capability for easy substitution of SW models (or parts of them) with HW components;
  - easy traceability of the simulator evolution by using standard tools as the CVS.
- Portability, to allow:
  - easy adaptability to the computer platforms used in the different project phases;
  - possible hosting under real time operating systems.
- Configurability, to allow:
  - full instantiation of a simulation through ASCII input files without any “hidden” hard coded data.
- Model Scheduling Reliability, to allow model execution “at the right time in the right sequence”. This is mandatory in order to avoid any unwanted delay at run time in case of multithread/multiprocessor configurations.
- Very light User interfaces for initialization and post-processing.

E2ES performance objectives can be expressed in terms of numerical error and execution time:

- The E2ES must have sufficient numerical precision to assess the impact of all relevant error sources.
- The execution time of the simulation of a given mission period must be always faster than the real time.

The kernel of NUMES is constituted as a collection of models representative of the satellite subsystem functions and of the P/L instruments. Of course, these models can be defined with different level of details, according to the design evolution and of the utilization exigencies of the simulator. As soon as new refined models are available, they can be easily included in the current version of the simulator (actually the simulator environment can be intended to be a sort of growing ‘repository’ of different models at different level of complexity). In this way the complexity of the simulator can grow from very slim configurations encompassing only some models exchanging data directly without any simulation of electrical interfaces, up to a full configuration capable to simulate the AOCS basic SW and the bus emulation. As soon as the HW units are available, the simulator can also migrate to a real-time context to allow possible performance verifications of the most critical units and functional tests of the overall system.
Model repository, libraries providing functions for I/O management, classical mathematical operation (vector, matrix, and quaternion), specific orbital functions (orbital parameters conversion, ephemeris computation, reference frames conversions) are foreseen as supporting tool.

The core of the E2ES is the scheduler, aimed at arranging in a proper sequence the methods belonging to the models constituting the simulator, and at supplying all the needed data to them.

The scheduler can considerably vary depending upon the system configuration to be simulated; nevertheless its general structure isn't expected to change. A scheduler performs:

- the preparation of the data necessary to setting up the execution of a particular entry of a model;
- the execution of an entry of a model depending on the position inside the execution flow (e.g. inside an integration step, at the beginning/end of an integration step or control step, just before or just after the computation of the control command etc.);
- the execution of particular simulator services as gathering all the output data needed to prepare the simulator output file.

Three different functions have been defined:

- Initialise: instantiation and the initialisation of the simulator;
- Execute: it arranges in sequence all the entries needed to run the models that constitute the simulator;
- Exit scheduler: releasing of the allocated memory and closing all the opened services.

Fig 2 represents the main data flux as managed by the Execute scheduler function:

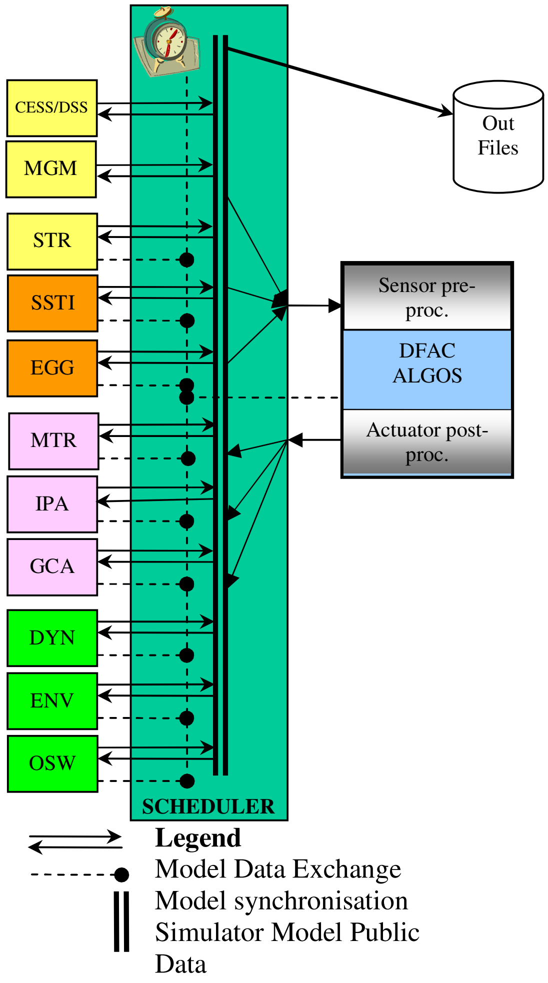


Fig.2 E2ES Scheduler Concept

Simulator Outputs Compared To Flight Data

Examples of outputs produced with the E2ES and compared with the actual in-flight behaviour of GOCE and of its Gradiometer are shown in Fig 3.

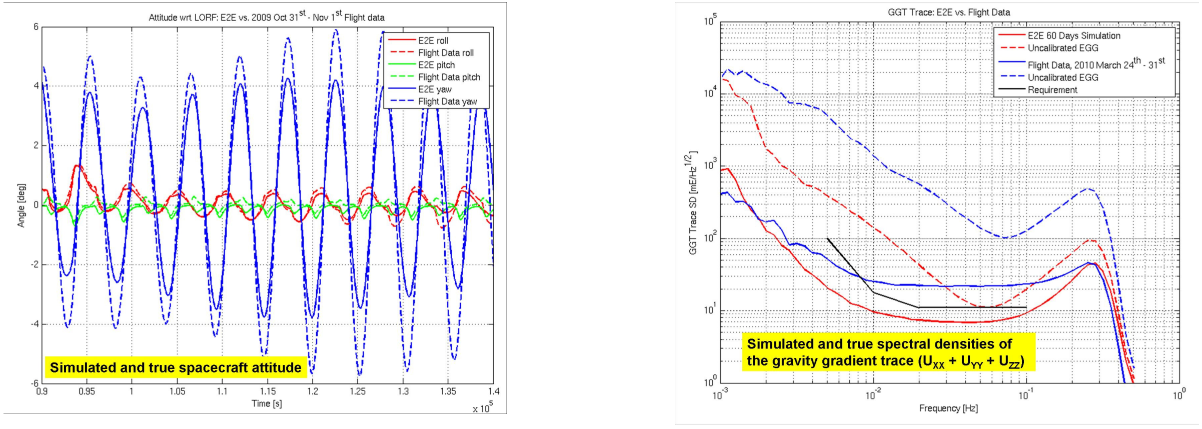


Fig. 3 Simulation Results against Flight Data

The comparison shows in general a good matching, consequence of the fidelity of the models of the environment, spacecraft and payload, and of the simulated orbit and attitude dynamics (including the drag-free and attitude control algorithms). An exception to this matching is constituted by spectral density of the gravity gradient components above ~30 mHz, due to an unexpected larger measurement noise, discovered only in flight and thus not modelled in the Simulator.

ON GOING PROJECTS: NGGM

Objective of the NGGM pre-phase A study is to establish a mission architecture concept aimed at the optimal recovery of the Earth time variable gravity field. The mission is based on a Satellite to Satellite laser Tracking (SST) observation technique and includes the analysis of a variable number of pairs of satellites with different orbits and inclinations. The NGGM E2ES is able to simulate:

- the identified mission scenarios including a multi-satellite dynamics;
- the defined control systems;
- the model of the various payload and sensor/actuator elements;
- the time-variable Earth gravity field model.

The NGGM simulator consists of two separate modules: a "forward module" developed by the Industrial Team, and a "backward module" developed by the Scientific Team. The former generates the satellite observables (i.e., inter-satellite distance variation and the non-gravitational acceleration measurements) starting from the signal produced by the variable gravity field at the satellite level. The second one recovers the variable gravity field from the satellite observables. Afterward the quality of the recovered gravity field is assessed and exploited to estimate parameters of target geophysical signals, in concert with the pre-defined scientific requirements. This way allows estimating the mission performances till to the early stage of the project.

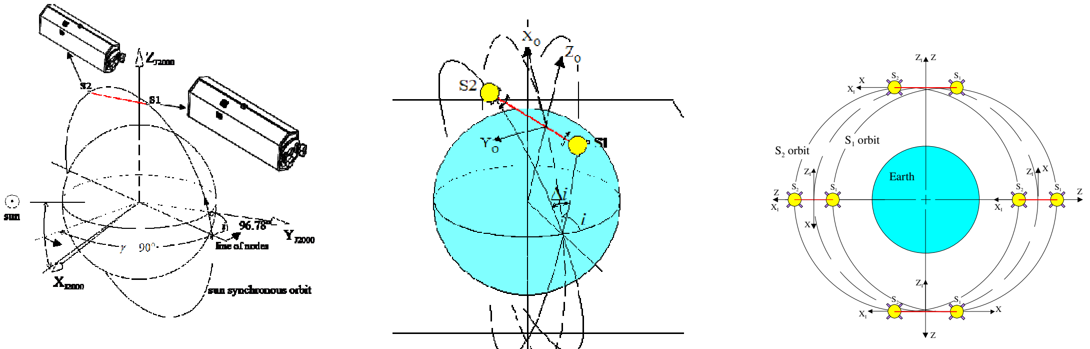
NGGM Mission Scenarios

The "forward module" is based on the NUMES frame requiring an extension of the original capabilities due to:

- the introduction of the multi-satellite dynamics;
- the more stringent numerical precision requirements (Numerical Dynamic range > 17 magnitude orders is mandatory due to the Typical orbital dimension of ~10^8 [m] & Accuracy of the Optical Metrology System ~10^{-9} [m]);
- the extended simulation time span required.

Fig. 4 sketches the analyzed satellite configurations.

In order to cope with these requirements, relying on the existent GOCE E2ES frame, the new developed models have been coded in C++ and the old ones modified in such a way to be accepted by the C++ compiler; so allowing for an easy multi-instantiation of a model eliminating every data clash risk.
The extended Numerical Dynamic range problem has been solved employing for the critical physical quantities the quadruple precision number format.



"in line"

"pendulum"

"cartwheel"

Fig. 4 NGGM Analysed Mission Configurations

Preliminary Results

Fig. 5 reports one of the outputs of the NGGM simulations representing the Relative Position of the Satellite2 wrt the Satellite 1 considering an in-line satellite configuration with the formation control active

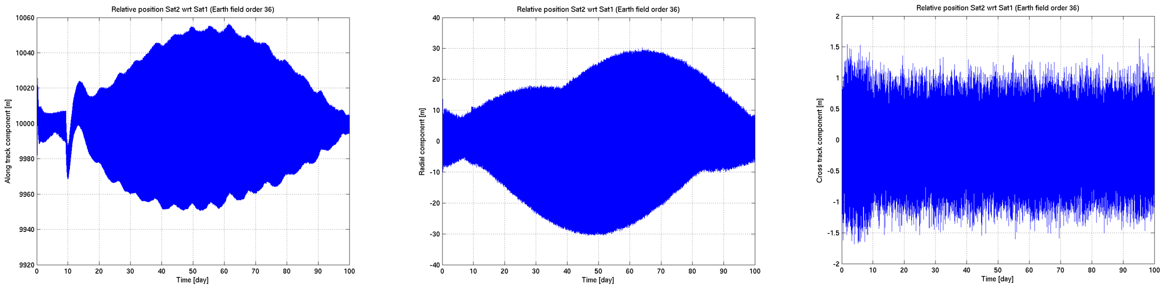


Fig.5 SatelliteRelative Position in an in-line configuration

ON GOING PROJECTS: EDL

The capability to perform a successful landing on Mars relies on a suitable design of the EDL (Entry Descent and Landing) system. It is therefore important to have a tool supporting the project across its entire lifecycle EDL starting from system design, analysis, and trade-offs up to the final Validation and Verification.

TAS-I is involved in several project requiring EDL design capabilities, i.e. the EDM module of Exomars and SAGE study. The objective of the EDL E2E Simulator is to answer to this need, providing an integrated environment supporting all the design and verification activities related to the EXOMARS EDL system and reusable for the others EDL projects. NUMES provides a suitable basic environment on which building the EDL E2E Simulator.
The EDL system requires the upgrade of the libraries with new models and new features.

Models

For simulating the atmospheric re-entry phase, an accurate modeling of atmosphere, winds and aerodynamics is of vital importance. These models are usually available in form of databases; FORTRAN code is made available for database management of the atmosphere models. The ad-hoc development of code for the management of the aerodynamics database has been instead required. These models have been built and integrated in the simulator with collaboration between aerodynamics experts and simulator developers. From the aerodynamics point of view particular effort has been spent in the definition of the transitions between the different atmosphere regions and relevant aerodynamics regimes (i.e. free-molecular flow, rarefied, continuous). From a simulator point of view the effort has been spent in the interface of these models guaranteeing flexibility in the selection of the options, parameters, thresholds and databases. This is due to the need, in the early phases of the project, of performing simulation with different interpolation laws, triggering thresholds, geometric data (shape and dimensions of the probe). An additional problem is related to the heavy computational effort required by these models, the atmospheric one in primis. The suitable scheduling of the model, the tailoring of the functions and the compilation options allowed reaching computational times compatible with the huge required number of simulations.

A further topic characterizing the EDL system is the parachute model. An accurate modeling of the parachutes sub-system requires ad-hoc complex multi-body simulator tools and very small integration steps to follow the high frequency phenomena related to the deployment phase (i.e. PASDA and/or DCAP-RT). In the End To End simulator it is required to have models with a minor fidelity level but compatible with less demanding integration steps.
The fidelity of the parachute model must grow in parallel to the project development and degree of fidelity of the simulator itself. In the early stage of the project, a simple mass less parachute model is used. More sophisticated models are currently under evaluation and shall replace the simplified model in the EDL simulator supporting the Exomars phase C/D.

Features

Multi-phase management

The simulation of the Entry Descent and Landing system requires the simulation of different Phases, each one identified by Events triggered by specific GNC functions. Each Phase is characterized by a different set of models, GNC algorithms, output to be monitored and often integration step.

The E2E simulator infrastructure has been then enriched with the capability of managing multiple simulation phases.

For each phase a Scheduler is defined, each one managing the execution of the relevant models. The integration step is defined Phase by Phase depending on the models requirement. For example, for Exomars the nominal integration step used is 0.01 (100hz); in the last phase the integration step is instead 0.001 (1000hz) driven by the thruster model requirements.

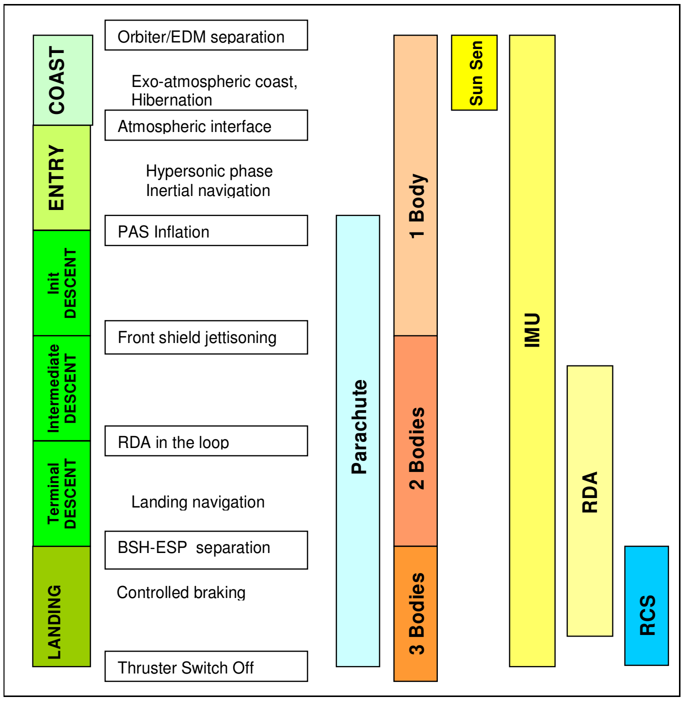


Fig. 6 EDL Exomars Phases and Relevant Active Models

The Mission Manager is a new element of the simulator added to manage the execution of the Schedulers.

All the models belonging to all the scheduler are instantiated at the beginning of the simulation (read of parameters from input files, object constructions, memory allocation, preliminary initialization). The mission manager drives the execution of the schedulers managing the transition from one scheduler to the next when a transition is commanded. The transitions are commanded when the GNC algorithms detects the relevant event triggers. In alternative, transitions can be commanded by input file as time tagged command. At the first call of each scheduler, the scheduler initialization function is executed; this function is in charge to initialize the new models (not initialized yet in the previous scheduler execution), to initialize the counters on the base of the scheduler integration step and to initialize the new state vector. Particular care is spent in the initialization of the position and velocity vectors, due to the fact that the CoM of the simulated bodies usually does not coincide with the CoM of the body simulated in the previous phase.

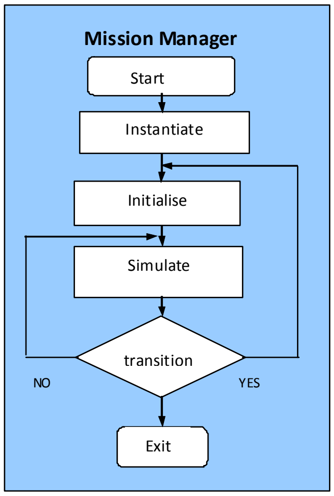


Fig. 7 Mission Manager Concept

Monte Carlo run Features

The EDL system and GNC design and verification requires an extensive use of Monte Carlo simulations. This capability has been achieved implementing a pre and post processing functions and an ad-hoc procedure,

The preprocessing is a Matlab procedures that, on the base on the default input files and driven by the user instruction relevant to the varying parameters of the simulation campaign (i.e. selected variables and their range of variability or statistical properties, number of runs), generates the input files for all the foreseen runs.

Then the Monte Carlo procedure manages the execution of the simulation and the storage of the output; this procedure is able to exploit the multi-processor architecture of the machine.

The stored outputs are then processed by an in-house developed tool for post-processing and statistical analysis

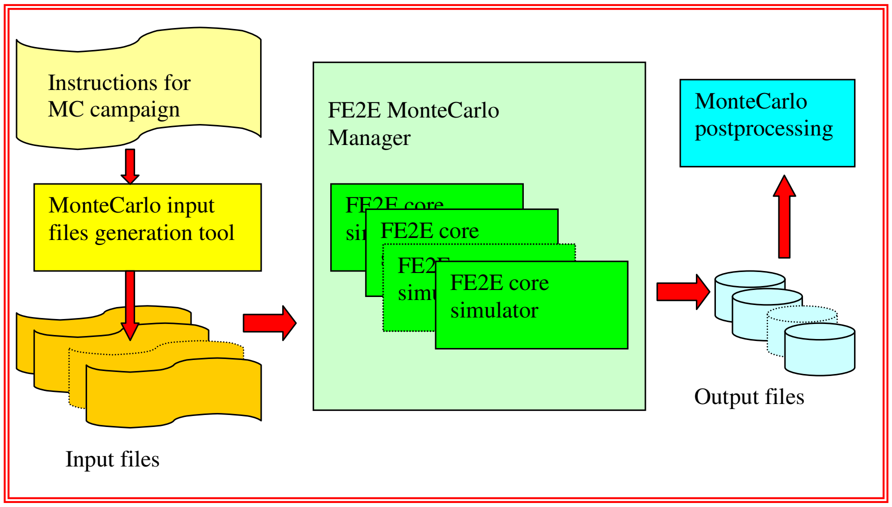


Fig. 8 Flow of a Monte Carlo Run

Preliminary Results

Fig.9 presents the typical results of a single run: altitude above planet surface, atmosphere density and angle of attack along the EDL trajectory, from EIP (Entry Interface Point) up to touch down,

Fig.10 presents the dispersion of the position at landing w.r.t. the ideal landing site results of a Monte Carlo simulation campaign.

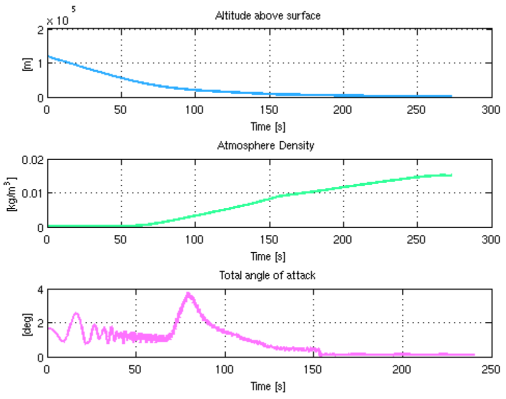


Fig. 9 Single EDL run results

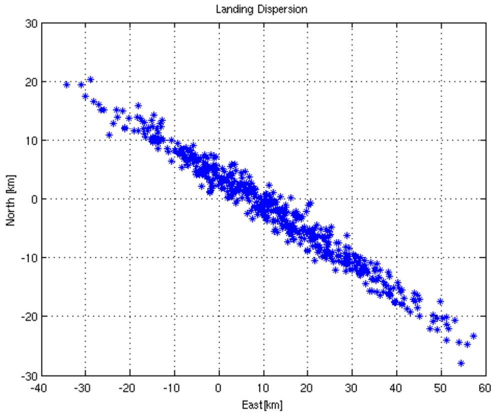


Fig. 10 Monte Carlo: Landing Dispersion

CONCLUSIONS

The basic assumptions on which NUMES relies on were:

- choice of high level compiled language C/C++ for the core simulator sep-up, coping with number crunching and numerical precision requirements
- choice of object oriented programming approach, coping with modularity and portability requirements
- use of mathematical tools like Octave/Matlab to support pre and post processing phases

The two different applications presented in this paper, the Gravimetric and Entry Descent and Landing Missions demonstrate the suitability of NUMES concept in supporting the Mission and GNC Design and Verification for the a wide range of applications. The upgrades of the framework to cope with the peculiarity of the two missions have been presented.

REFERENCES

[1] G. Catastini, S.Cesare , De Sanctis S., M. Dumontel, M. Parisch, G. Sechi "Predictions of the GOCE in-flight Performances with the End-to-End System Simulator" *In Proc. 3rd International GOCE User Workshop, ESA SP-627 (CD-ROM), ESA Publications Division, European Space Agency*. Noordwijk, The Netherlands.

[2] G. Catastini, S. De Sanctis, M. Dumontel, M. Parisch "The E2E System GOCE Simulator" *DASIA 2006*.

[3] G. Catastini, De Sanctis S., M. Dumontel, M. Parisch "Model Scaling Approach for the Goce End to End Simulator" *DASIA 2007*.

[4] G. Catastini, S. Cesare , M. Dumontel, M. Parisch, G. Sechi, R. Floberghagen "The End to End System Simulator of Goce" *ESA Living Planet Symposium* 28 June - 2 July 2010, Bergen, Norway.

[5] System Support to Laser Interferometry Tracking Technology Development for Gravity field monitoring "*Summary Report*" *SD-RP-AI-0589* December 2008.