AN INTEGRATED FRAMEWORK FOR RECONFIGURABLE MISSION PERFORMANCE AND FUNCTIONAL ENGINEERING SIMULATION TOOLS

J. Araújo(1), T. Hormigo(1), F. Câmara(1)

(1)Spin.Works Lda.
Rua Rodrigues Sampaio, 97, 4º, 1150-279 Lisboa Portugal
Email: joao.araujo@spinworks.pt

INTRODUCTION

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 Guidance, Navigation and Control (GNC) algorithms. A simulation template consisting of a generic decomposition and interconnection of simulator blocks, a uniform functional and data flow, and consolidated coding standards have 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. A model library has been set up, and software tools to quickly deploy new simulation tools were developed.

This simulation framework has been successfully implemented in the scope of European Space Agency (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). The 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.

A SIMULATION TOOLS FOR SPACE SYSTEMS

Engineering simulation tools have become a key component in the development of modern space systems, regardless of mission, mission phase, environment, vehicle, or system/subsystem. The increasing specialization of companies, as well as the widespread availability of simulation software, tend to devolve into an array of simulation tools, developed at each contractor site, to support the design and testing of each component under their responsibility, requiring additional work to ensure compatibility with other simulation platforms at consecutively higher system integration levels.

While specific requirements are demanded by each project and system type, the development of dedicated models, the use of environment models at various fidelity levels, and time constraints in project execution, frequently result in sub-optimal solutions from the standpoint of their future applicability, many of the underlying physical processes involved are common even between very dissimilar projects ([1],[2]), and a conscious choice is often made by companies and institutions to work towards developing simulation tools where both the development, testing and validation efforts are minimized across tasks, projects, and entire fields of activity.

It is with this perspective that Spin.Works has developed its' own simulation framework. Simulation tools based on this framework are applicable to different mission profiles, environments and vehicles of different types, configurations and complexity. The framework shall also support a range of foreseeable evolutions with negligible adaptations. The following aspects have been considered as minimum requirements for simulation tools based on the framework:

- **Missions**: initially targeted towards all phases of Entry Descent and Landing, Rendezvous and Docking, Formation Flying and Unmanned Aerial Vehicle missions
- **Degres-Of-Freedom (DOF)**: including but not limited to 3DOF, 4DOF, 6DOF simulations
- **Fidelity**: models of varying fidelity, user-selectable during the simulator configuration
- **Vehicle**: initially rigid-body models, but capable of supporting multibody and/or flexible body simulations
- **Planetary bodies**: as a minimum Earth, Mars and the Moon, but easily adaptable to any bodies including small irregularly-shaped bodies
- **Monte-Carlo capability**: fully supported
- **Real-Time**: initially supported via IP communications only.

This paper described the properties, capabilities and some of the recent applications of Spin.Works' simulation framework.

In part B, the baseline architecture of simulation tools in the scope of this framework is described. This architecture contains three components. First, we describe a generic simulation template (part B.1), consisting of a system decomposition, data flow architecture and coding standards. A second component of the framework architecture is a model library (part B.2), which contains both a model repository (where all relevant model data is stored) as well as a model and parameter database relying on XML files. The third component of the framework is a set of software tools which interact with the other two components to produce simulation tools compliant with the simulation template. This is described on part B.3.

The application of this framework to two specific examples related to ESA projects is then described in part C, showing the main aspects of the framework in terms of the minimization of the effort involved in the deployment of engineering simulation tools.

B SIMULATION FRAMEWORK ARCHITECTURE

The simulation framework hereby described consists of three components: a **simulation template** applicable to a broad array of simulation tools; a **model library** in which a set of validated models are stored, in addition to the necessary code to operate these models in the scope of a simulation tool; and finally the **software tools** which use data within and between the two elements to quickly operationalise simulation tools.

B.1. Simulation Template

The **simulation template** consists of a **standard system decomposition** and interlinking (shown in **Fig. 1**), a **standard functional and data flow architecture**, and **consolidated coding standards** (derived from [3]) applicable to all simulation software. This template can be seen as a generic architecture integrating the acquired knowledge of the company with respect to the basic operations and desired features of dynamic simulation tools involving GNC systems.

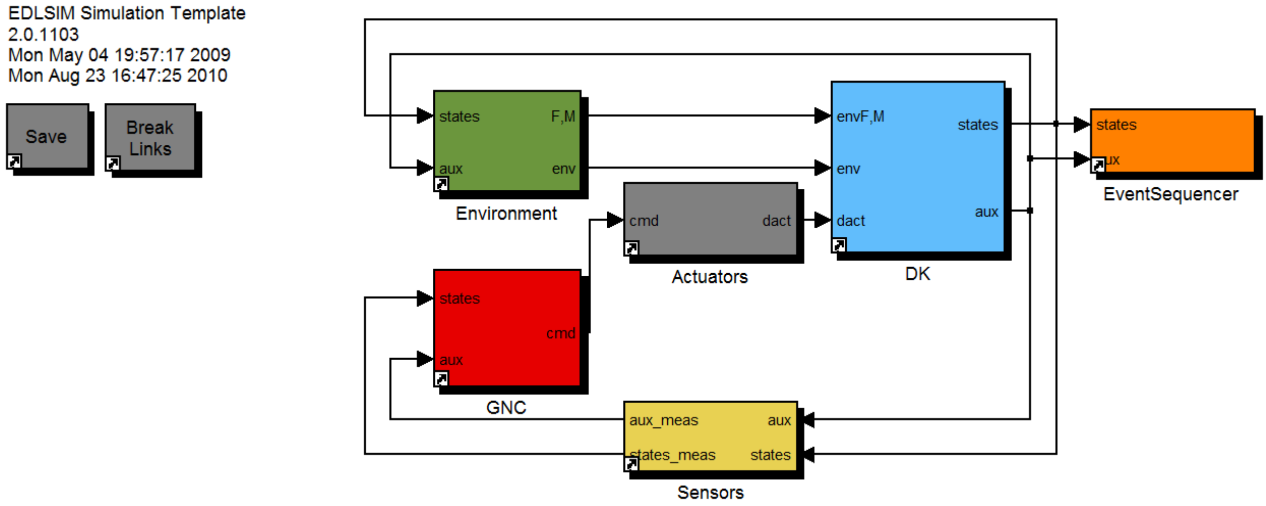

Fig. 1. Generic system decomposition and interlinking of a simulation tool. Example taken from the simulation tool used in an ESA project.

The functional and data flow architecture established for the development of simulation tools can be decomposed into 3 main elements: the **engine**, the **core** and the **interface**. These elements can be described as follows:

- **Interface**: This element consists of the Man-Machine Interface (MMI) and external interfaces. The user interface is composed, on one hand, of eXtensible Markup Language (XML) files and text editors for editing simulation configuration files, and on the other hand the Matlab command-line through which the user can set-up the simulation functional mode, run simulations, make use of post-processing functions, etc.
- **Engine**: this is an intermediate layer of the simulation tool, which works as an interface/translator between the user and the simulation core. It is responsible for assigning values to parameters, setting up and configuring the simulation core; for managing the simulation (single or multiple) execution; and for post-processing and plotting the simulation results.

- **Core**: this is the innermost layer of the simulation tool, where the main simulation processes occur, i.e. numerical integration. It comprises the implementation of the mathematical representations of on-board system models (e.g. sensors, actuators, GNC) and of the Real World.

Taking into account these three elements, the sequence of operations during the execution of a simulation tool can be described as follows: first, an MMI is operated by a user to modify simulation parameters and trigger the execution of a Simulation Manager. This Simulation Manager then calls a Configuration Module in order to load system-related parameters from the available input files, designate the simulation model to be used, and select the state propagation methods. Once relevant data has been loaded into the workspace, a simulation is executed. Once execution is completed, the time history of system states, as well as a list of relevant parameters, is saved to a time-tagged binary file for post-processing.
The interaction between the elements of **Fig. 1** is further depicted in **Fig. 2**.

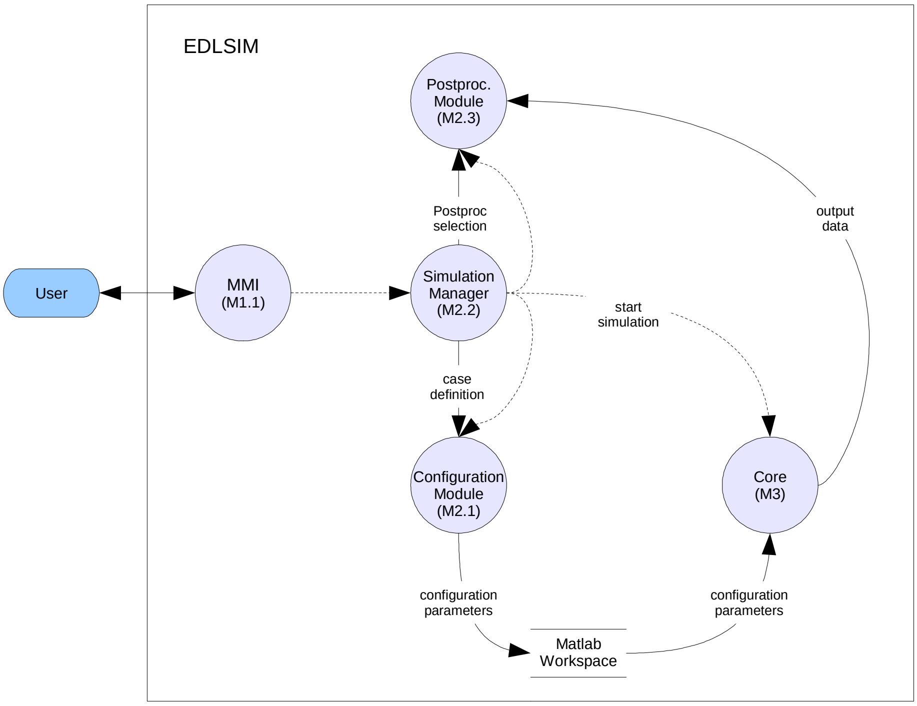

Fig. 2. Simulator data and control flow diagram (1st level)

B.2. Model Library

A **model library** is used within the Spin.Works framework, similarly to [1], to promote the use of standard models and parameters across simulation tools. The model library consists of two elements: a centralized **model repository**, containing both the models validated in previous simulation tools and any software components associated to each model, and a **model and parameter database** consisting of two XML files, populated with metadata from the available library models.
Within the model repository, each model component (source code with compiling setup data, configuration code and binary files, run-time executable code and test routines) is organised according to a standard folder structure, similar to the structure used in each simulation tool. A separate, fixed hierarchy is also imposed to organise the repository itself. Currently this hierarchy contains the following elements:

- **Environment** - planetary atmospheric models (Earth and Mars atmospheres are currently implemented, and include wind models), planetary shape and gravity models (including ellipsoidal models, spherical harmonics-based models for the Earth, the Moon and Mars, and 3rd-body gravitational models based on planetary ephemerides databases), topographic models, etc.
- **Vehicle** - core dynamics and kinematic models used by simulation tools; vehicle-specific force and moment models:
    - **Equations of Motion** - vertical-plane motion equations, 3DOF translational motion equations both in inertial frames and planet-fixed around a spherical body, 3DOF quaternion-based angular motion equations, and aerodynamic-angles kinematics equations.

    - **Aerodynamic models** - currently contains two capsule models (Huygens-like and Apollo-like models) and an enhanced lifting body model (HL-20)
    - **Solar Radiation pressure models** - generic model based on simple vehicle geometry parameters.
- **GNC** - consists of an expanding set of Guidance and Control models including:
    - Entry guidance, Terminal Area Energy Management guidance, Approach and Landing guidance, terminal descent guidance
    - Flight control systems applicable to entry capsules, lifting bodies, space planes and powered landers
    - Several image processing models applicable to data collected by camera and Lidar sensors
    - Hazard detection and avoidance models based on camera sensors, Lidar sensors and as part of hybrid (Camera+Lidar) systems
- **Sensors** - IMU sensors (accelerometres and gyroscopes), air data sensors, radar altimeter and doppler velocity sensors, magnetometers, hazard detection sensors (Camera, Lidar)
- **Actuators** - thruster models and aerodynamic surface actuator models
- **Toolbox** - frame transformations (100+), flight data computations and parameter conversions, as well as special mathematical functions and interpolations.

In parallel to this model repository, a Matlab-accessible database was set up where information on each block is stored.
This database consists of **two files** (one for the **models**, and one for the **parameters** in the repository). The model-related file holds model descriptions, a model ID within the library, inputs/outputs and the parameters used, along with typical configurations and a list of model elements used in its operation. The parameter-related file holds information on each parameter used in simulation tools: parameter description and units, a parameter ID within the library, and typical parameter values and configurations.

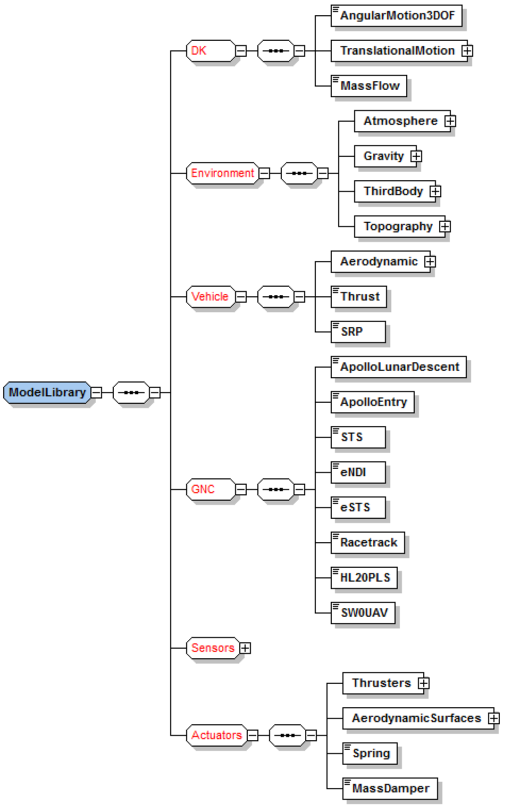

Fig. 3. Partial schematic representation of XML file containing metadata on all library models.

B.3. Quick Deployment of Simulation Tools

Based upon the available model repository and the information contained on the XML database, software has been developed to assist in the automated configuration of the initial aggregation and deployment of a simulation tool.
This software is founded on database queries, including logical and relational operations leading to the creation of simulation tool-specific databases containing relevant information on the contents of each model in a simulation tool.

Simulation tool aggregation and deployment is led by a specialized user. This user drags individual components from the model library and drops them onto a clean sheet simulator. These actions trigger successive database queries in order to identify the necessary parameters to be configured for the execution of a simulation, with configuration files being modified according to the real-time updated list of simulation parameters. In this way, a fully operational, and initially basic tool, quickly becomes operational, based on previously developed and validated models.

Once this initial process is completed, all links to the model library are removed and the simulation tool becomes an independent identity. At this point the user typically proceeds to the tailoring of the simulation tool and the development of specialized models needed in the particular scope of the project. This approach allows a user to expedite the deployment of a simulation tool, proceeding directly to more productive activities such as GNC development tasks.

C SIMULATION FRAMEWORK APPLICATIONS

Spin.Works has been involved in ESA activities related to the Entry, Descent and Landing (EDL) phases of a space mission, where development efforts have been focused towards the implementation and integration of components for EDL simulation tools, in support of GNC development tasks. This is the case for two examples which are presented in this section, one related to the development of high-performance Hazard Detection and Avoidance (HDA) algorithms applicable to the powered landing phase on the lunar South Pole (ESA NEXT-LL Phase A project), and the other related to the development of innovative guidance algorithms applicable to the Terminal Area Energy Management (TAEM) phase of capsules, lifting bodies and space planes (ESA TAEM project).

C.1. Simulation Tools for the Next-Lunar Lander Phase A

The objective of the overall NEXT-LL Phase A was to design a spacecraft to safely land on the region of the lunar south pole. Within the scope of the project was the identification of hazard detection and avoidance strategies to ensure a safe landing in these regions of large shadows and dramatic terrain. Spin.Works' responsibility in NEXT-LL was to trade-off and select sensors and algorithms towards fulfilling these strategies, given a required success probability, known/expected terrain properties, alternative descent trajectories, and the available space-qualified hardware.

With these responsibilities in mind, two simulation tools and a range of off-line computation tools were developed. The first simulation tool was a Mission Performance Simulator (MPS)-type tool to study the interactions between trajectory, observation geometry and mission timeline for the descent phase (from 5km range to touchdown). Motion was then assumed to occur strictly on an inertial frame and with a constant gravity. Two variations of an Apollo-like powered descent guidance were used. Generic sensor properties (field-of-view, nr. of pixels) were used to obtain time histories of imaging ground resolution and variability of the resolution within each image, depending on observation geometry. Thrust level limitations and throttling control depth data were modelled, based on the preliminary selection of thrusters for the mission. Loss of fuel mass was modelled based on the known specific impulse *Isp* of the available thrusters.

This simulation tool was extensively used to identify how the descent trajectory affects hazard detection performance. Other analytic and statistical methods to determine this performance, for different observation geometries, were used to identify suitable descent trajectories and imaging sensors. An initial picture of which variables have most influence on the hazard detection performance was compiled and documented, serving as a baseline for subsequent studies.

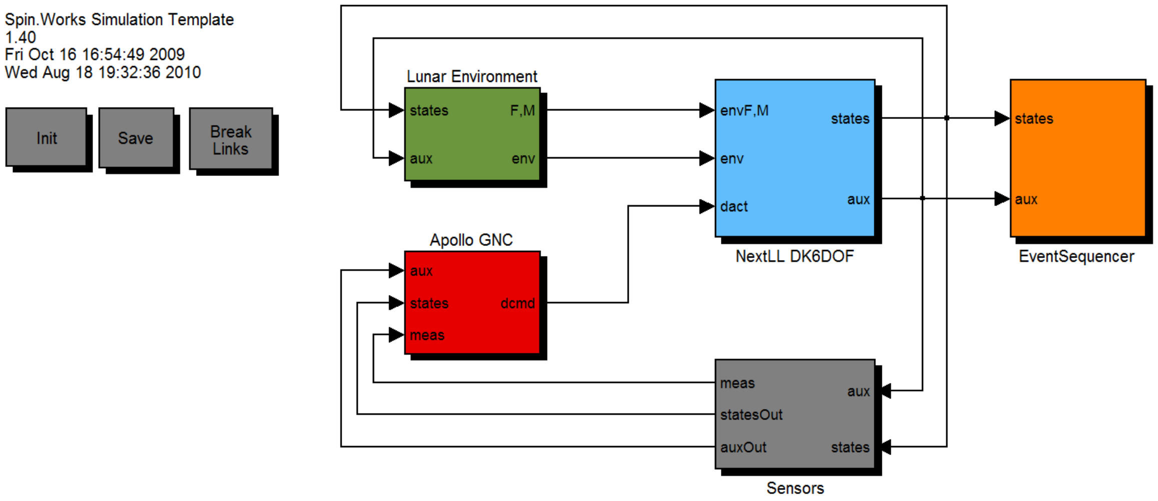

Fig. 4. 6DOF Simulation Tool developed for the ESA NEXT-LL project

The second tool, akin to a Functional Engineering Simulator, was built in support of a consolidated mission design, integrating a large set of relevant models in the scope of hazard detection and avoidance: realistic terrain models, sensors, actuators, full 6DOF guidance and control, and hazard detection and avoidance algorithms.

This simulation tool also assumed that the translational motion took place in an inertial frame, but added the 3DOF angular motion, with the kinematic equations based on attitude quaternions. To the powered descent guidance algorithms used before, a reference attitude generation module and a nonlinear attitude control algorithm were added, so as to accurately simulate the lander motion throughout the descent phase. Camera and Lidar sensor models were implemented, as well as a widely available terrain generation and visualization tool[1], enabling the capture of more realistic imaging data for the verification of hazard detection algorithms. While thruster limitations were inherited from the previous simulation tool, more thrusters had to be included for attitude control purposes, namely requiring a thrust assignment unit. Thrust levels at each individual thruster were added up and aggregated by thruster type, with the resulting mass flow calculated according to differences in *Isp* between the different types of thrusters used.

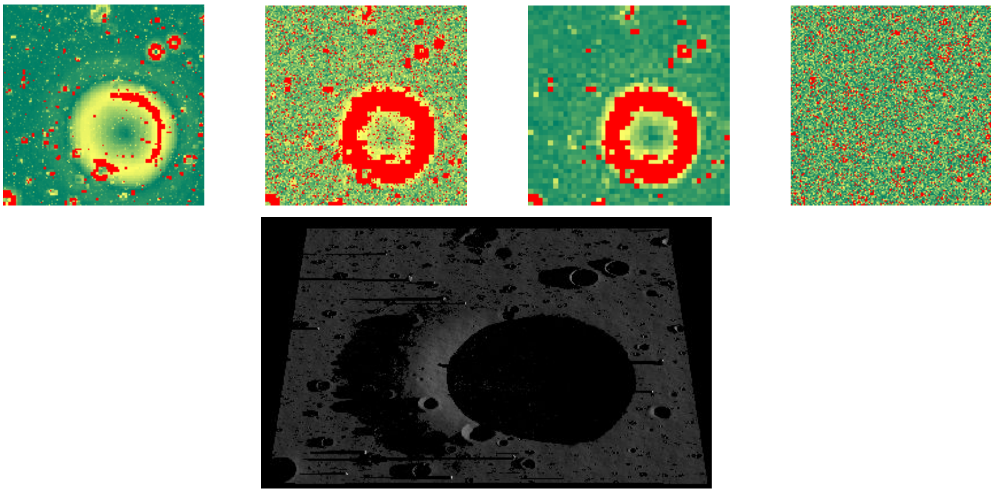

Fig. 5. Shadow, texture, slope and roughness maps obtained along a descent trajectory towards the lunar surface, and projected to a regular grid (top), and camera image of the same terrain (bottom).

Several off-line tools were developed, allowing a thorough performance evaluation for the hazard detection and avoidance system. However, this simulation tool was deemed essential for the purpose of validating the overall approach taken for HDA during descent, since it required the individual validation and the integration of realistic models from nearly every relevant subsystem involved in the detection and safe landing site selection during the descent trajectory. Still, the outputs of this simulator were not directly used during the course of the project; instead, the results became instead a portion of the validation process for the descent strategy finally adopted in the Phase A of the NEXT-LL project (selection of sensors and hazard detection algorithms, taking the descent trajectory profile as a reference input), while they also served to strengthen the confidence in the soundness and validity of the simulation tool itself.

C.2. Simulation Tools for the TAEM Project

The ESA project "New Guidance Schemes for the Terminal Area Energy Management phase", led by Spin.Works in partnership with ASTOS GmbH, was aimed at the implementation of innovative guidance systems applied to the entry phase of low lift-to-drag vehicles (capsules), and the TAEM phase of lifting bodies and space planes. The project entailed the identification of leading TAEM guidance schemes and a comparison, via large-scale simulations, between these guidance schemes and new guidance approaches.

The TAEM project involved the application of simulation tools to a wider set of vehicles and GNC systems than the NEXT-LL project. However, in this project, both the environment, the baseline dynamics (with the exception of a 6DOF simulation tool required for landing simulations) and many of the required parameter conversions were essentially the same. This particular fact encouraged the implementation of a single simulation tool applicable to all the vehicles and GNC systems, and a small library was implemented in separate, allowing for the selection of test

---

[1] The camera and lidar models used for the ESA NEXT-LL project are made available through the Planetary and Asteroid Natural scene Generation Utility (PANGU), developed by the University of Dundee.

procedures, mission profiles, environment models, dynamics models, GNC models and vehicle models and configurations, via the simple modification of model tags in the header portion in the main simulator configuration file.

Fig. 6. Model Library for the 4DOF/6DOF simulation tools used in the ESA TAEM project.

In accordance with this approach, a 4DOF simulation tool was initially deployed to support the implementation of GNC algorithms for the different vehicles. Translational equations of motion in a planet-fixed frame about a spherical central body were used. The Earth was modelled as the WGS-84 ellipsoid, and a J2-based gravity model was used. The NRLMSISE-00 and standard US76/US62 atmosphere models were used, as well as the HWM-93 horizontal wind model and a parametric wind model to mimic the wind magnitude variation along the atmospheric boundary layer.

No sensor or navigation model was used except the Apollo navigation model, tightly connected to the Apollo Entry Guidance. The modelled vehicles were a sub-scale version of the Apollo capsule and two configurations of the HL-20 enhanced lifting-body. The guidance schemes implemented for low L/D vehicles were the Apollo Guidance and an Enhanced Nonlinear Dynamic Inversion entry guidance ([6]), and for high L/D vehicles, the Space Shuttle STS guidance algorithm, the Racetrack algorithm developed at the IFR Institute of Stuttgart University and the Enhanced STS ([5]) algorithms (an extension to the latter algorithm was introduced in order to enable 4DOF landing simulations).

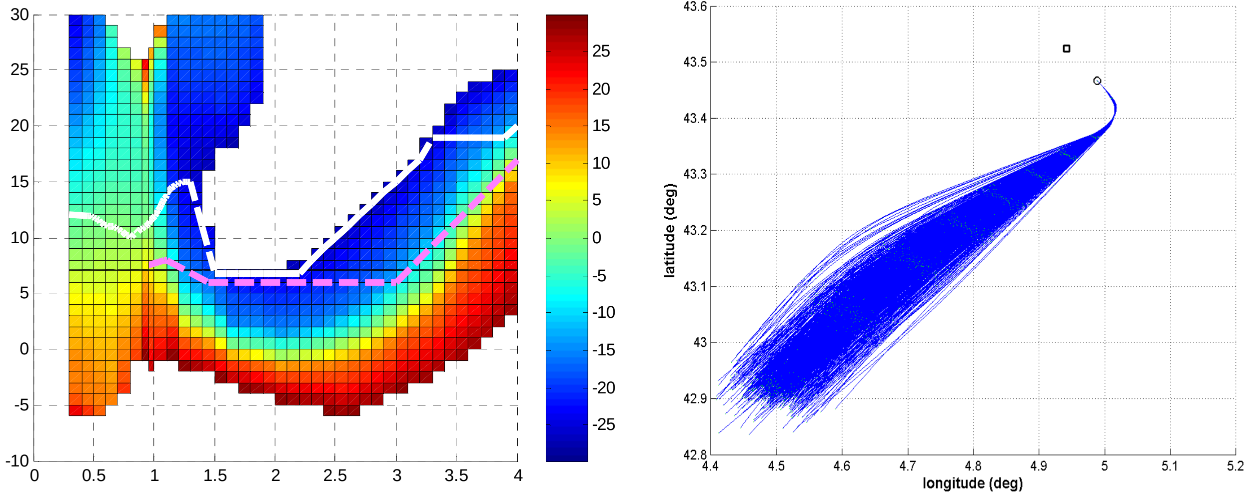

Fig. 7. HL-20 trim elevator deflections (left, in degrees) as a function of Mach number and angle-of-attack (x and y-axis), including allowable region (white) and nominal profile (pink). Results from a 4DOF Monte-Carlo test campaign for an enhanced Space Shuttle-type TAEM guidance algorithm (right).

The 4DOF simulation tool also included aerodynamic models in the form of trimmed aerodynamic force coefficient tables. These tables were obtained, for the Apollo-like capsule, at the angle-of-attack which results in equilibrium at each Mach number. For the HL-20 vehicle the trimmed aerodynamic coefficients were calculated for each combination of Mach number, angle-of-attack, speed brake and center-of-gravity position, yielding 4-D tables for the lift and drag coefficients, and longitudinal trim surface deflections were computed in the form of pseudo-elevator deflections (these deflections correspond to deflections of the available vehicle aerodynamic surfaces according to a surface mixer logic implemented by NASA during the HL-20 PLS studies in the early 1990's).

The 6DOF simulation tool was an evolution of the 4DOF simulation tool, applied to a space plane configuration to simulate the flight of the HL-20 vehicle from the Terminal Entry Point until touchdown. For this purpose the equations for the 3DOF angular motion were added, as well as the GNC system (extension of guidance schemes to touchdown, and implementation of two flight control systems - the original HL-20 G&C system as in [4] and an internally-developed flight control system -, implementation of a control surface mixing logic), and actuator models (the original HL-20 actuator models were implemented). All other models were kept unchanged from the 4DOF simulations.

The above simulation tools were validated against an (external) reference simulation tool, and then used in the scope of Monte-Carlo simulations, whereby the guidance schemes were tested under realistic mission scenarios and parameter variations. The uncertainties used included atmospheric and wind uncertainties in magnitude and direction, aerodynamic and mass property uncertainties, and perturbations to initial conditions. Stringent requirements were attached to the relevant parameters (such as parachute deployment precision, landing precision, maximum dynamic pressure and g-loads, velocity vector accuracy at key trajectory points, etc.), leading to a substantial verification and validation of each guidance scheme as well as a comparative evaluation between the different algorithms implemented within the simulation tools.

The battery of tests performed using these accurate dynamic models, realistic mission profiles, and alternative GNC systems, ultimately resulted in the demonstration of key advantages between the new TAEM approaches and the current guidance state-of-the-art, clearing the path for further incremental development and, eventually, real-world testing of safe and precise terminal area energy management guidance schemes.

D CONCLUSIONS

This paper described the framework developed by Spin.Works for the purpose of implementing space system simulation tools. This framework consists of a simulation template, which comprises a standard system decomposition, a data flow architecture and coding standards applicable to all simulation tools; a model library which contains the company's consolidated models as well as a database implemented as two XML files which contain metadata related to the models and the parameters involved in simulation tools, and software tools which allow a systematic manipulation of both the model library and individual simulation tools to promote the expeditious deployment and the exploitation of space system simulation tools.

Concrete applications of this simulation framework in the context of two recent ESA activities are discussed, including the role played by these simulation tools within the scope of the tasks under the responsibility of Spin.Works, the required simulation components, and the results obtained through the use of these simulation tools.

Other recent activities by the company involving the use of this framework, namely in projects related to the development of guidance systems for Mars precision landing, navigation algorithms and integrated navigation sensors, and of GNC algorithms for unmanned aerial vehicles, have in the meantime extended the experience of the company with this framework, and further hinted of significant productivity gains achievable through the repeated use of a single, flexible, generic dynamic simulation tool concept for a broad set of GNC-related applications.

REFERENCES

[1] E. St. John-Olcayto. G. Johns, A. Pidgeon, and C. Philippe, "EAGLE: An Extensible, End-to-End Simulation and Evaluation Framework for Planetary EDLS", 7th *International Planetary Probe Workshop*, Barcelona, June 2010.

[2] S. A. Striepe, D. W. Way and A. M. Dwyer, "Mars Science Laboratory Simulations for Entry, Descent, and Landing", *Journal of Spacecraft and Rockets*, vol. 43, no. 2, pp. 311-323, March-April 2006.

[3] ESA Requirements and Standards Division, "Space Engineering - System Modelling and Simulation", ECSS-E-TM-10-21A, 16 April 2010.

[4] E. Bruce Jackson, Christopher I. Cruz and W. A. Ragsdale, "Real-Time Simulation Model of the HL-20 Lifting Body", NASA-TM-107580, July 1992.

[5] F. Câmara, "Development and Design of the Terminal Area Energy Management Guidance for a Reusable Launch Vehicle", Master of Science Thesis, TU-Delft, The Netherlands, 2003.

[6] T. Hormigo, J. Araújo and F. Câmara, "Nonlinear Dynamic Inversion-based Guidance and Control for a Pinpoint Mars Entry", AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, August 2008.