6th International Conference on Astrodynamics Tools and Techniques/ March 2016

RAPID DEPLOYMENT OF DESIGN ENVIRONMENT FOR EUCLID AOCS DESIGN

Aerospace

F. Cacciatore, R. Sánchez, A. Agenjo, N. Puente, C. Ardura, L. Olier, V. Gómez, M. Saponara, G. Saavedra

www.sener.es

Introduction

- **Euclid is a cosmology mission dedicated to study the geometry and the nature of the Dark Universe with unprecedented accuracy**
  - Observe a 15000 deg^2 wide area of the sky from L2
  - Scientific goals result in very demanding performances for the AOCS subsystem: RPE < 75 mas over 700 sec during observations
- **Euclid S/C: procured by ESA and supplied by TAS-I**
- **SENER is the prime contractor of the AOCS sub-system**
  - Work executed in partnership with ADS-NL
  - SENER responsibility is for the whole AOCS. Design, implementation and verification of GNC/AOCS modes and their SW, units subcontracts and overall verification.
- **Current AOCS status is fully in line with expectations**
  - AOCS SRR completed in 1 month and AOCS PDR in 7 additional months
  - PDR colocation was held at the end of November
  - Units' subcontractors activities are being launched with good progress

Euclid Design Simulator and AOCS Development

- **In the frame of EUCLID AOCS development two simulators are employed:**
  - ESE: Engineering Simulation Environment, responsibility of Elecnor-Deimos
    - *Functional Engineering Simulator, one of the formal test environments supporting overall Euclid AOCS verification and validation process*
  - EDS: Euclid Design Simulator
    - *The AOCS design environment*
- **Euclid Design Simulator**
  - Developed during Phase-B2 activities, will be evolved and maintained over the whole project life-cycle
  - Environment employed in design iterations: it's the framework in which AOCS modes, functions, interfaces and databases are being created
  - Units models: initial models in EDS → evolution through suppliers info → revert to EDS for update and test → specification

SENERIC

- **SENERIC is a suite for high-fidelity tools and models for the design, analysis, and validation of an AOCS**
  - Initially funded by Spanish Ministry of Technology, turned in internal R&D project for continuation
- **SENERIC is at the core of DKE, mathematical functions, sensors, and actuators of EDS. Models and libraries available for:**
  - AOCS Equipment: sensors and actuators
  - AOCS/GNC Functions: for rapid prototyping, analysis and design
  - Environment: force-torque perturbations, ephemerides, time references
  - Mathematical operations & transformations
  - S/C Dynamics and kinematics
- **Employed in the design of Planck and IXV AOCS, in PROBA-3, and for the design and auto-coding of OPTOS Attitude Control System**
- **Math library taken as basis of EUCLID AOCS math library**

Dynamics

- **Dynamics: joint integration of the S/C and reaction wheels, sloshing, and telescope Filter Wheel Assembly (FWA/GWA) dynamics equations**

$$\begin{bmatrix} I_{tot} & I_{RW}U_{RW} \\ I_{RW}U_{RW}^{T} & I_{RW}I_{n\times n} \end{bmatrix} \begin{pmatrix} \dot{\bar{\omega}}_{SC} \\ \dot{\bar{\omega}}_{RW} \end{pmatrix} = \begin{pmatrix} -\bar{\omega}_{SC} \times (I_{tot}\bar{\omega}_{SC} + I_{RW}U_{RW}\bar{\omega}_{RW} + \bar{u}_{FWA}I_{FWA}\omega_{FWA}) \\ -\bar{T}_{f}(\bar{\omega}_{RW}) \end{pmatrix} + \begin{pmatrix} \bar{T}_{ext} \\ \bar{T}_{RW} \end{pmatrix} + \begin{pmatrix} -\bar{u}_{FWA}I_{FWA}\dot{\omega}_{FWA} \\ \bar{0} \end{pmatrix}$$

- **The FWA/GWA is modelled by explicitly imposing a kinematic motion by directly injecting the angular rate and acceleration profiles**
  - Acceleration obtained respecting integral relation with rate
  - This ensures a correct torque profile and conservation of angular momentum

Simulation Core - Actuators

- **MPS (Micro-Propulsion System). Cold gas thrusters for attitude control in scientific observations.**
  - Accurate model needed in line with strict pointing requirements
  - Modelling of thrust noise frequency profile by filtering a white noise source
- **RCS (Reaction Control System). Employed for attitude and orbit control**
  - Effects included: integration & sampling, thrust time profile (rise+tail), noise, plume impingement
  - RCS pulse forced scaled over the AOCS cycle to model correct linear impulse independently of the sample time
- **RWA (Reaction Wheels Assembly)**
  - "Distributed" model: separate integration of equation of motions and simulation of dynamic behaviour
  - The RWA will have to be stopped and restarted hundred of thousands of times during the mission
  - Wheels braking modelling must ensure conservation of angular momentum
  - Dead time included during torque reversals
  - RWA dynamics simulates static+viscous+Coulomb friction

Simulation Core - Sensors (I)

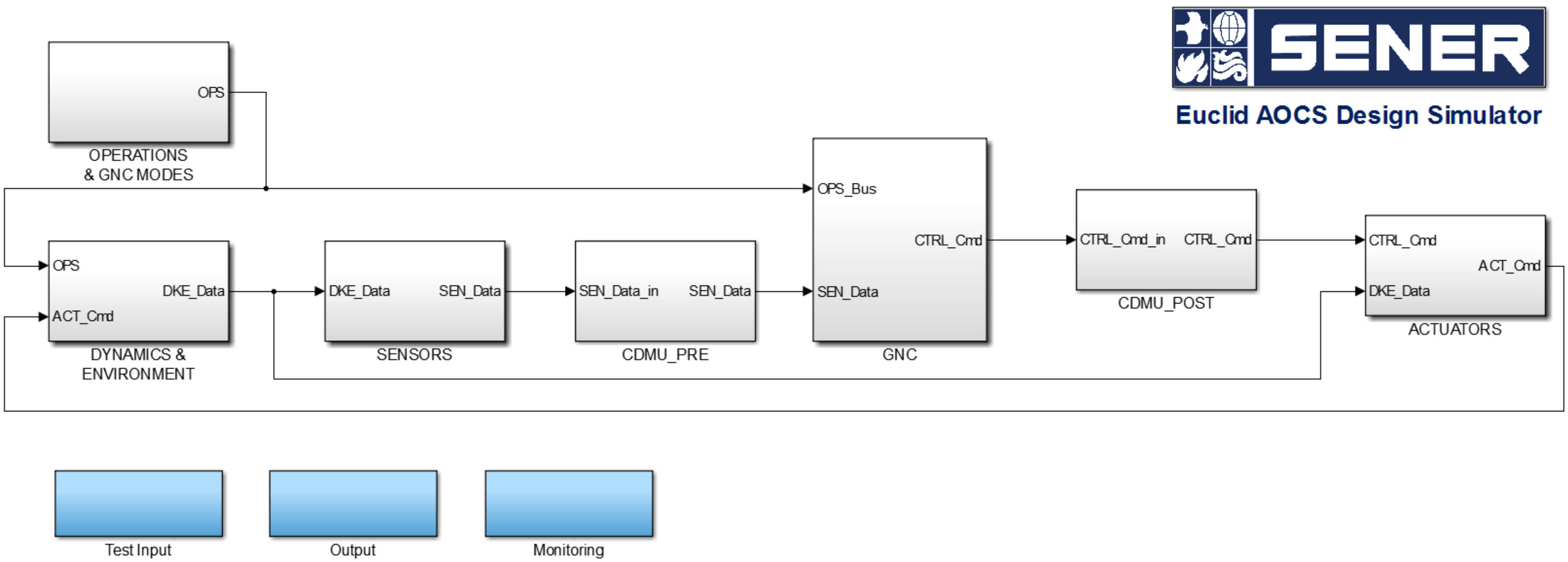
- **Gyroscope**
  - Rate integrating gyro model employed for 4 independent heads
  - Ad hoc filtering of White noise sources employed to reconstruct Allan variance profile
- **Accelerometers**
  - Approach similar to gyro.
  - Linear accelerations due to sensors accomodation + rotational dynamics included
- **Coarse Rate Sensors**
  - Analogous to gyros
- **Sun Sensors (SAS)**
  - Non-ideal cosine output simulated
  - Albedo included based on offline computation taking into account Earth illumination and sensor FOV

Simulation Core - Sensors (II)

- **FGS (Fine Guidance Sensor)**
  - CFI for the AOCS; physically mounted on telescope
  - Most relevant sensor for science observation phase
  - Complex functioning: sensor modes logic, attitude lock, acquisition delays, data outages, measurement delays, absolute/relative measurement, performances function of angular velocity
  - Custom in-house model built implementing specified behavious and characteristics
  - Wrapping built to allow EDS to swap in-house model for customer-provided model
  - Customer-provided model incrusted in EDS with mínimum modifications
- **Star trackers**
  - Integration+process delay taken into account
  - Velocity-dependent noise included
  - Spatial effects due to FOV and detector characteristics (pixel error) modelled by first and second order transfer functions with variable (velocity and rotation axis dependent) gains

Simulation Core

- **CDMU (Central Data Management Unit)**
  - AOCS is executed as part of the onboard software, running on the CDMU
  - CDMU pre-GNC and post-GNC functional subsystems included in model in order to represent correctly the timely data flow due to the finite AOCS execution time

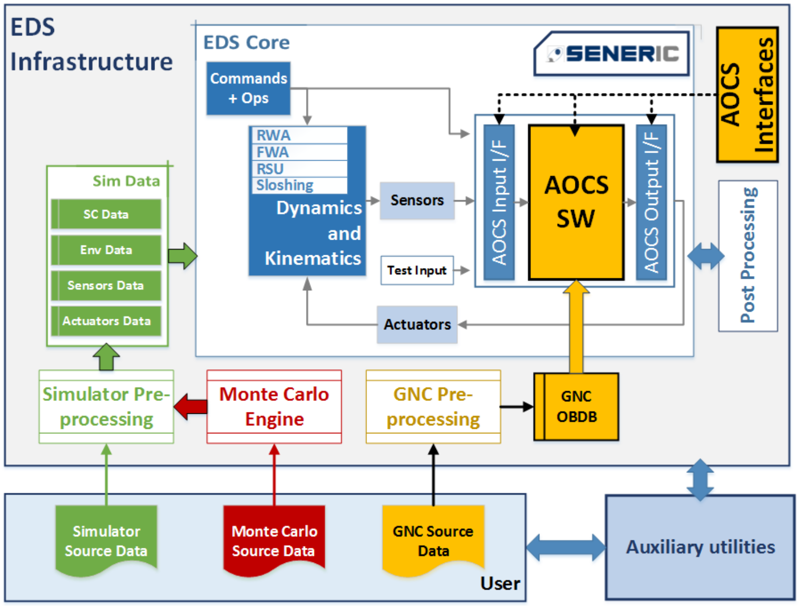


EDS Workflow and Infrastructure

- **The EDS was developed taking into account the following usage guidelines:**
  - fast input/parameters set-up
  - Allow quick algorithms design loops: modify AOCS algorithm or unit model, run test case, check output, modify again
  - Allow systematic execution: manage large data sets, run and store in an orderly fashion different sim cases, including Monte Carlos
- **EDS high level components:**
  - Simulink simulation core
  - MATLAB simulation and data management infrastructure
  - Utilities for workflow efficiency and ancillary operations

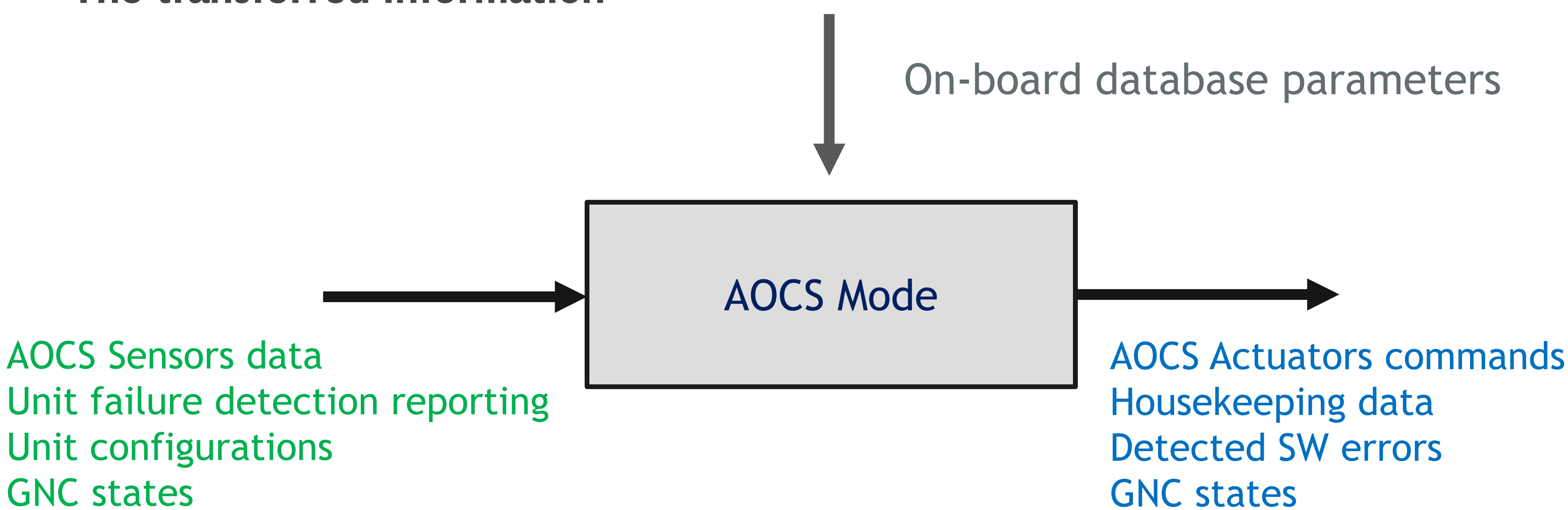
Simulations execution

- **Three execution modes allowed:**
  - Design Run
    - *allow rapid iteration on algorithms design and simulator parameters, minimizing the set-up effort.*
    - *aim: configure the simulation with a single MATLAB command*
  - Systematic Run
  - Monte Carlo
- **Data management**
  - Source databases: user-defined parameters
  - Pre-processing: transforming user-friendly parameters into data needed by simulation core
  - XML databases and m-files used: m-files for quick setup and debug, xml for more structured runs
  - XML files can be generated directly from m-files, to reduce data preparation effort



AOCS Interfaces (I)

- **Interfaces definition plays a relevant role in the design and development process**
- **All AOCS modes except Safe will be auto-coded**
- **The AOCS modes will have software I/F only with AASW-MAN (manually coded AOCS software). Logic interfaces are with**
  - Sensors and actuators through the AASW-MAN
  - Directly with AASW-MAN itself
- **The transferred information**



AOCS Interfaces (II)

- **Strong data typing approach is employed in the definition of the AOCS modes models**
- **Input/Output AOCS-AUTO data are organized into Simulink Buses, analogous to C data structures**
  - The Bus Objects are autocoded into C-code type definitions, which constitute part of the actual flight software
- **The AOCS interfaces are defined in terms of such data buses and their elements, with their data types and dimensions**
- **The interfaces must be iterated and documented, and will be needed by the AOCS-MAN SW and by the ESE to embed the AOCS-AUTO code**
- **EDS contains a utility that allows generating the interfaces minimizing the iteration effort**
  - Interface signals, data types and sizes are defined in a specific Simulink model
  - Resulting Bus Objects for all modes are automatically generated (single command)
  - The Bus definition libraries revert in the Simulator interface blocks, ensuring consistency with what expected by GNC

AOCS Software Development (I)

- **The AOCS is being developed according to the Model-Based Design (MBD) philosophy**
  - The model is an executable specification that is continually refined throughout the development process
- **MATLAB/Simulink allow to**
  - Link models with requirements
  - Generate C code with comments and tags that trace with the model and the requirements
- **AOCS algorithms:**
  - Part of the model to be included in the AOCS Application Software
  - Are being evolved in the EDS
  - Will eventually be developed as auto-generated code on the embedded processor
  - Will be verified in SIL, SVF and HIL facilities

AOCS Software Development (II)

- **Model-Based design development and validation plan follows the evolutionary model**
  - Iterative approach where progressive versions of the model/software are planned
  - For each version a model validation is foreseen
- **The AOCS model will evolve based on simulation loops performed in the EDS environment**
  - Simulations on the EDS will be the start point to develop the formal validation tests
- **Each AOCS mode is created as a referenced model**
  - Root class representing the overall mode
  - Decomposition in smaller elements according to functionalities (submodes, functions) and assignment to specific C code functions during autocoding
  - Simulink architecture defined and analysed taking into account Simulink to C mapping

The way to see the future

Thank you