

PRECISE EMULATION OF COMMANDS FOR ESA INTERPLANETARY MISSIONS

Livio Tucci⁽¹⁾, Joachim Keil⁽²⁾, Vicente Companys⁽³⁾, Sara De La Fuente⁽⁴⁾

⁽¹⁾ Terma GmbH c/o European Space Operations Centre (ESOC), OPS-GFI, Robert Bosch Strasse 5, 64293 Darmstadt, Germany, E-mail: Livio.Tucci@esa.int

⁽²⁾ EDS c/o ESOC, OPS-GFI, Robert Bosch Strasse. 5, 64293 Darmstadt, Germany, E-mail: Joachim.Keil@esa.int

⁽³⁾ ESA c/o ESOC, OPS-GFI, Robert Bosch Strasse 5, 64293 Darmstadt, Germany, E-mail: Vicente.Companys@esa.int

⁽⁴⁾ AIRBUS, Spain E-mail: Sara.delaFuente@airbus.com

ABSTRACT

This paper describes the architecture and the main features of the command emulators developed to support the Mars Express (MEX), Smart-1 and Rosetta missions within the Flight Dynamics (FD) Division at the European Space Operations Centre (ESOC).

FD emulators are used to precisely model the spacecraft (S/C) dynamics and the behaviour of the Attitude and Orbit Control and Measurement System (AOCMS) of the S/C in order to:

- Validate operational telecommands sent to the S/C
- Generate high precision telemetry test data for FD internal system tests
- Troubleshoot AOCMS anomalies and perform AOCMS analyses and studies

The architecture and wide functionality of the FD emulators are detailed, describing how the actual software (s/w) is able to meet the needs. Commonalities and differences between different mission specific implementations are enlightened.

The different fields of application of emulators within FD are illustrated.

Finally, as an example of application, emulation results are shown, corresponding to the Rosetta asteroid fly-by mode (AFBM) test that will take place around the first Earth swing-by.

Keywords: emulation, S/C dynamics, interplanetary missions, telecommands generation, end-to-end consistency check, telemetry generation, asteroid fly-by

1. EMULATOR ARCHITECTURE

1.1 Overview and Requirements

ESOC FD is currently supporting the operations of three ESA interplanetary missions: MEX, Smart-1 and Rosetta.

For each of these missions a S/C emulator has been developed, as a part of the FD system preparation. The requirements that shall be met by the FD emulator are the following:

- It shall provide high precision modelling of S/C dynamics, environment (e.g. ephemerides, orbit, forces, torques), AOCMS, sensors and actuators dynamics.
- It shall be fully integrated in the FD system being able to correctly interface with existing FD FORTRAN libraries and data formats.
- It shall provide high simulation speed.
- It shall be possible to check and validate operational telecommands created by the FD Command Generation subsystem.
- It shall be capable to generate simulated test telemetry data for testing of the FD system and for training of the FD Mission Team.
- It shall provide capabilities to troubleshoot anomalies of the AOCMS of the S/C and perform analyses of the S/C AOCMS performance.

1.2 Common Features

The major components of the emulators are:

1. AOCMS model
2. Dynamics models
 - Attitude dynamics model
 - External torques model
3. Sensors and actuators models
4. Ground/Space interfaces models

FD emulators provide the following interfaces for integration in the FD system:

- On the input side they are able to automatically process AOCMS telecommands both in accelerated and real time applications. More in detail, command parameters sequences generated by the Command Generation subsystem, which need to be emulated, are translated in a form readable by the

emulator in an automatic way allowing emulation capability of very long telecommands sequences.

- On the output side they are able to format relevant AOCMS engineering quantities in the form of telemetry packets compatible with the operational TM definition of the S/C.

Different approaches have been followed for the implementation of the Rosetta/MEX emulators and the Smart-1 emulator. Details on the different approaches will be given in chapter 1.3.

For what concerns the attitude dynamics model, Euler's equations for a central rigid body with rigid articulations (i.e. solar arrays, articulated antennas) have been modelled. In Smart-1 also solar arrays flexibility has been taken into account.

Lander ejection dynamics (Beagle for MEX, Philae for Rosetta) have been considered as impulsive manoeuvres with angular momentum exchange.

The following external torques have been included in the attitude dynamics model:

- Gravity gradient torque (all 3 missions)
- Solar radiation pressure torque (all 3 missions)
- Aerodynamic torque (Smart-1)
- Residual S/C dipole magnetic torque (Smart-1)
- EP engine longitudinal torque (Smart-1)

Sensors and actuators have been modelled according to manufacturer documentation. In some cases pseudo code for sensors and actuators algorithms was available in the documentation and has therefore been fully used in the implementation.

FD emulators do not perform propagation of the S/C orbital state, but just read it from an orbit file provided by the manoeuvre optimisation subsystem.

JPL ephemerides for Sun and planets are used, to e.g. evaluate:

- Solar radiation pressure torque
- Gravity gradient torque
- Sun incidence on specific S/C surfaces (i.e. solar arrays, sun acquisition sensors, radiators etc.)

In MEX and Rosetta emulators a star catalogue including magnitudes and inertial directions of 3227 stars is used in order to model star tracker measurements.

Two kinds of databases are used:

- AOCMS s/w parameters database
- Flight Dynamics database (FDDB) which contains all kinds of physical quantities characterising the S/C, its sensors and its actuators (e.g. S/C inertia

tensors, surfaces solar reflectivities, thrusters specific impulse).

1.3 Two Different Implementation Approaches

Even if the general architecture is the same as described in the previous chapter, the specific implementation of FD emulators has followed two very different approaches:

For Rosetta and MEX, emulators have been coded in FORTRAN 77 using as reference manufacturer documentation (e.g. AOCMS s/w requirements document). On-board s/w is represented by functional emulation (i.e. AOCMS s/w requirements have been used to achieve a representation of the s/w).

For Smart-1 the S/C manufacturer (Swedish Space Corporation, SSC) initially provided FD with their AOCMS and S/C dynamics emulator, which had the following characteristics:

- Coded in Matlab/Simulink on a PC machine under Windows Operating System (OS)
- Contains the AOCMS core of the real flying onboard s/w
- Developed by the manufacturer to internally test S/C AOCMS algorithms

SSC s/w has been enhanced, modified and ported (PC/Windows -> Solaris/Unix) in order to finally obtain a stand-alone C-executable program, fully integrated as an operational tool into the FD system on Solaris/Unix environment.

This new approach has allowed FD to drastically shorten s/w development time and to achieve very high representativity of the system since the emulator incorporates real AOCMS s/w of the S/C.

Therefore, in the case of Smart-1 mission, the emulator design was focused mainly on:

- Enhancing and tailoring SSC Smart-1 s/w to specific FD needs
- Resolving portability issues
- Resolving interface issues with already existing FD FORTRAN infrastructure
- Creating I/O interfaces:
 - Automatic processing of command sequences
 - Telemetry data generation

A bit in more detail, the Smart-1 emulator development steps can be summarised as follows:

- Simulink model enhancement (PC/Win)
- Simulink model modification to create topological I/O interface (PC/Win)
- Automatic C source code and Make file generation using Matlab Real Time Workshop (PC/Win)

- Porting of the C source code and Make file into the Solaris/Unix environment
- Model I/O interface with FD system s/w development (Solaris/Unix)
- Automatic processing of command sequences s/w development (Solaris/Unix)
- Cross compiling of the C source code (Solaris/Unix)
- Final generation of a C-executable stand-alone application (Solaris/Unix)

2. MODES AND FIELDS OF APPLICATION

The emulators have three main fields of applications:

- AOCMS Telecommands Validation
- Test Telemetry Data Generation
- AOCMS Contingencies Troubleshooting and Analyses

Emulators can run in two different modes:

- Batch, accelerated mode
- On-line, real time mode

In batch mode all the telecommands required to be sent to the S/C in order to run a specific scenario are available at the beginning of the simulation and are loaded in the simulation timeline at initialisation.

In on-line mode the user can interactively send commands to the emulator. Emulators are run in on-line mode for FD team training, allowing for intervention in the course of emulation in a timely realistic manner.

2.1 AOCMS Telecommands Validation

AOCMS telecommands validation via emulation is used routinely in FD operations for MEX and Rosetta. It is performed in batch mode. The effects of the telecommands are systematically checked by the emulator before they are sent to the mission control system for up-link.

Telecommands validation is achieved via consistency checks between quantities predicted by the command generator and the corresponding emulated quantities.

As an example, the evaluation of the S/C attitude depointing between the commanded attitude (ground guidance) and the emulated attitude validates that a certain number of working assumptions used in the telecommands generation process sum up to an acceptable attitude depointing value. Some of the sources of mismatch between the ground-commanded attitude and the emulated attitude can be identified as follows:

- Approximation of ground-commanded attitude via Chebyshev polynomials
- Different reaction wheel torque evolution

- Attitude sensors misalignment and measurement noise

2.2 Test Telemetry Data Generation

Generating test telemetry data has the very important objective to train FD Mission Team in internal system tests starting approximately six months before S/C launch.

Before the ESOC wide simulation campaign starts, FD performs a series of internal tests covering different representative mission scenarios in order to validate the internal interfaces between the different FD subsystems and to be sure to be in a fully ready status to support S/C operations.

The complete chain of FD operational activities is tested: emulated tracking data are given to Orbit Determination subsystem, which performs the orbit determination. Based on the knowledge of the S/C state vector at a given epoch Manoeuvre Optimisation subsystem computes manoeuvres to be commanded in order to achieve the desired S/C trajectory. Manoeuvres command parameters and ground-commanded guidance parameters are computed by Command Generation subsystem and fed to the emulator, which emulates the scenario, providing real time telemetry data to Attitude Monitoring subsystem which monitors S/C performance as a consequence of the up linked commands. Telemetry data are generated exactly in the standard format as expected by Attitude Monitoring subsystem, which can therefore check the correct functioning of their analysis tools.

Interactions between Attitude Monitoring and Command generation subsystems take place as in real operations: not only the S/C dynamics but also in a wider sense the FD operations chain is properly emulated. The emulator closes the “loop” in the FD systems tests, allowing S/C AOCMS performance assessment as a result of up linked telecommands and FD Mission Team training.

Nominal and non-nominal scenarios (e.g. S/C anomalies and failures) can be emulated.

2.3 AOCMS Contingencies Troubleshooting and Analyses

Emulators can be used in operations to troubleshoot true S/C contingencies and also to predict S/C performance in critical manoeuvres.

If an AOCMS subsystem anomaly or failure is reported via telemetry by non-nominal values of some parameters, hypotheses on the origin of the problem can be made and validated with “ad hoc” emulator runs. The emulator is a very important tool to support decision-

making in critical operations scenarios e.g. the Launch and Early Operations Phase, MEX Mars Orbit Insertion.

Finally the emulators can also be effectively used as general AOCMS off-line analysis tools, being able to predict the performance of critical S/C activities.

The following emulation scenario will illustrate this latter concept in detail.

3. ROSETTA ASTEROID-FLY-BY MODE CHECKOUT

In September 2008 Rosetta will fly by the asteroid Steins and in July 2010 the asteroid Lutetia. Given the uncertainties of the orbital parameters of the asteroids, it is not possible to program on-board a predefined attitude profile which ensures pointing of the asteroid during the fly-by. For this reason, the AOCMS of the S/C provides the Asteroid Fly-By Mode (AFBM), which allows for closed loop tracking of the asteroid based on navigation camera (NAVCAM) measurements.

The AFBM is of high complexity and it has been proposed to check it out in flight before it is operationally used for the first asteroid. FD has searched for opportunities for an AFBM checkout. The main drivers for such checkout are that an extended object is needed and that the relative motion of the S/C with respect to this object shall resemble (as far as possible) that of the asteroid fly-by.

An opportunity has been found for performing the AFBM checkout by tracking the Earth Moon short after the first Earth Swing-by (March 2005).

3.1 Background of the asteroid-fly-by mode

There are six quantities that determine the fly-by scenario:

- The relative position vector from the asteroid to the S/C at time t_0
- The time derivative of it, the vector of relative velocity

The gravity forces between asteroid and S/C are negligible. Because of the relatively small distance between the two objects during the whole fly-by scenario Sun's gravity vanishes in the same locally inertial frame. This can be expressed as follows:

$$\frac{d\vec{v}_{rel}}{dt} = 0 \quad (1)$$

The velocity vectors of the S/C and of the asteroid, and hereby also the relative velocity vector are known very accurately from Orbit Determination. This well-known

constant is commanded at the transition to AFBM. The on-board representation of the remaining fly-by variables is chosen under these perspectives:

- The relative distance is not relevant for the attitude guidance
- In a deterministic model the uncertainty of the relative position vector at the time of the transition to AFBM leads to the largest attitude error at the time of closest approach (being the moment of maximum scientific output)

Consequently the fly-by variables are:

- The inclination i of the true fly by plane with respect to an inertially fixed reference plane (e.g. the predicted fly-by plane)
- The angular separation α between the relative position and the relative velocity vector

The equations of motion for these quantities are necessary for an on-board propagation that takes into account corrective measurements from the NAVCAM. These can easily be derived:

The fly-by plane is invariant leading to:

$$\frac{di}{dt} = 0 \quad (2)$$

For α the equation can be derived from Fig. 1:

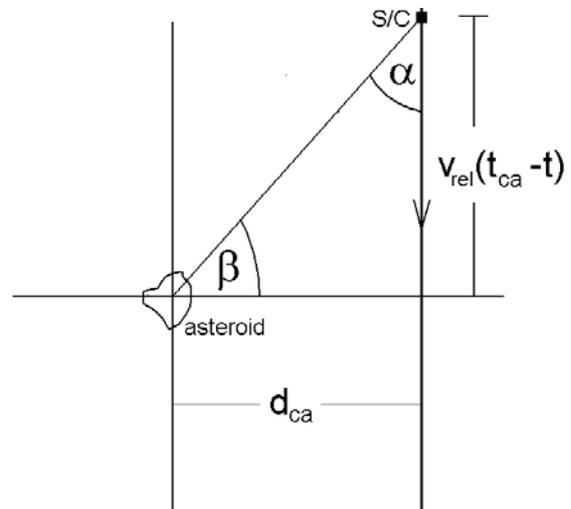


Fig. 1: Fly-by geometry

$$\beta(t) = \arctan\left(\frac{v_{rel}(t_{ca} - t)}{d_{ca}}\right) \quad (3)$$

With: v_{rel} : magnitude of the relative velocity
 t_{ca} : time of closest approach
 d_{ca} : magnitude of distance vector

From

$$\alpha = \frac{\pi}{2} - \beta \quad (4)$$

It follows:

$$\alpha(t) = \frac{\pi}{2} + \arctan\left(\frac{v_{rel}(t-t_{ca})}{d_{ca}}\right) \quad (5)$$

The evolution of α with time determines the parameters v_{rel}/d_{ca} and t_{ca} .

The equations of motion for α can now be formulated by differentiation of equation (5):

$$\frac{d^2\alpha}{dt^2} = \frac{2\left(\frac{d\alpha}{dt}\right)^2}{\tan(\alpha)} \quad (6)$$

The quotient of d_{ca} and v_{rel} is for both asteroids of the order of 200 sec. This means that the angle that the distance vector traverses between the times 200 sec before and 200 sec after the closest approach is 90 deg. The one-way light times are about 20 min for the Steins fly-by and about 25 min for the Lutetia fly-by. This shows the strong need for an autonomous guidance that keeps the asteroid in the field of view of the NAVCAM.

3.2 The guidance Kalman filter

As a starting point for AFBM the S/C is put in such an attitude that the asteroid is in the centre of the NAVCAM. The telecommand for the actual transition to AFBM contains initial values for equations (2) and (6). The propagation of the state

$$\begin{pmatrix} \alpha \\ \frac{d\alpha}{dt} \\ i \end{pmatrix}$$

is done in a Kalman filter. Every 2.5 sec measurements of the NAVCAM correct the propagated state and adjust the theoretical trajectory to the true one by an innovation of the state vector. This innovation is computed in relation to the guidance predicted by means of equations (2) and (6) for the time of the corrective measurement. In the period between two measurements the prediction equations are used for the guidance. The applicability of these equations only depends on the correctness of the initial estimate of the relative velocity unit vector. However, the Kalman filter estimator would only fail if the innovations would diverge. They need not necessarily converge to zero.

3.3 Asteroid Fly-by Mode Emulation

The emulator can be used to assess the behaviour of the asteroid fly-by mode. A realistic scenario can be constructed, for which noise on the measurements of the NAVCAM as well as on the reaction wheel torques, the gyros and the star tracker measurements are considered. The impact of the disturbance torques originating from the movement of the solar arrays on the control performance can be assessed. The effect of a change in the command parameters can be systematically investigated.

In particular, for the AFBM checkout with the Moon the emulator has provided confidence that the AOCMS can cope with a scenario, which is approximate, but not fully equivalent to the true asteroid fly-bys, e.g.:

- The approximation of the relative velocity as a constant is not as accurate for the Moon case as for the asteroid fly-by.
- The plane of relative motion, which for practical effects could be considered constant for the asteroid fly-by, changes for the Moon case considerably.
- The motion of the relative direction vector is much slower than for the asteroid case, resulting in much smaller rates.

The reason is that the locally inertial frames of the Moon and the S/C are not identical due to the different distances to the Earth being close.

An extensive simulation campaign has been conducted with the emulator to validate that the AFBM checkout can be performed safely with the S/C and that it can fulfil the objective of the checkout, i.e. to provide overall confidence on the AFBM. This campaign has also demonstrated that the approach for computing command parameters is adequate and that the system reacts to the commands as expected.

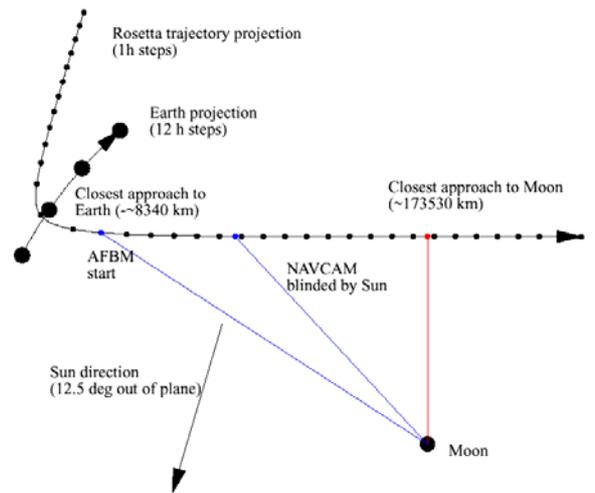


Fig. 2: Geometry at first Earth swing-by in a Moon fixed frame

Another fact of the Moon fly-by geometry is that the Sun gets close to the NAVCAM FoV long before the closest approach and blinds the NAVCAM (see Fig. 2). NAVCAM measurements will be interrupted about 9 hours prior to closest approach, when the Sun exclusion angle becomes 60 deg. From that point onwards the prediction equations (2) and (6) are the only source for the attitude guidance. As discussed above the conditions in which these equations are valid are not perfectly met.

The emulator allows predicting the evolution of the off pointing of the NAVCAM axis from the Moon, which reaches a value of about 1.25 deg at the end of the asteroid fly-by mode one hour after closest approach. The field of view of the NAVCAM is 5 deg in diameter. Therefore the Moon is at any time in the FoV of the optical instruments during the asteroid fly-by mode.