NEW SIMULATION CONCEPTS APPLIED TO FORMATION FLYING TESTBED DESIGN

Rachid Atori [(1)], Dominique Baudoux [(1)], Dominique Torette [(1)], Alastair Pidgeon [(2)], Francesco Pace [(3)], Raffaella Franco [(4)]

[(1)] Spacebel, I. Vandammestraat 7, 1560 Hoeilaart, Belgium
[(2)] SciSys UK Ltd, Methuen Park, Chippenham, Wiltshire, SN14 0GB, UK
[(3)] GMV Aerospace and Defence S.A., Isaac Newton 11 Tres Cantos E-28760 Madrid, Spain
[(4)] ESTEC, Keplerlaan 1, 2200 AG Noordwijk, The Netherlands

KEYWORDS

Simulator, SMP2, testbed, model, portability, formation flying, PROBA-3.

ABSTRACT

This paper presents the evolutive Formation Flying TestBed (FFTB) developed in the context of the PROBA-3 mission preparation.
The PROBA-3 mission is aimed to demonstrate in orbit the formation flying technology and to ensure the parallel development and validation of the ground verification tools and facilities. PROBA-3 is therefore the FFTB primary target, although the testbed objective is to set up an environment that can be tailored to support other future formation flying missions.
The main requirement for the infrastructure is to support the different phases of the PROBA-3 project activities;

- the early concept validation in a functional engineering simulation context,
- the flight software development and testing in a Software Validation Facility,
- the system AIT activities with EGSE and hardware in the loop capabilities,
- the ground segment preparation and validation with an operational simulator.

This paper presents the two dimensional scope of the testbed; demonstrating on one side the life cycle utilization scenarios, and on the other side the formation flying specific aspects. This is followed by the description of the main innovative architectural concepts allowing the scaling of the testbed functionality along the different setups and the model portability architecture. The concepts are demonstrated through formation flying demonstration cases.

1. TESTBED MAIN CHARACTERISTICS

FF and PROBA-3 Specific Requirements

The need for test facilities to support the project lifecycle is neither new nor related to formation flying. However, its application to innovative formation flying missions implies new requirements induced by the technical and organisational specificities of such missions.
Even if PROBA-3 is the reference demonstration application of the FF testbed, the solution must address a priori any formation flying mission. In particular, the following characteristics have been taken into consideration:

- The testbed must support the parallel simulation of a set of satellites, with a correct time synchronisation and with real-time performance as a minimum. Two satellites are foreseen in the case of PROBA-3, but additional pseudo-satellites could be entirely simulated by a dedicated processor in the spacecraft and provide a cradle for experimental FF algorithms.
- The spacecrafts in the formation are coupled, either loosely or tightly, to achieve the function of one single virtual spacecraft. This requires specific distributed algorithms (Guidance, Navigation and Control functions, FDIR, Formation Flight Management…), specific sensors and specific Inter Satellite communication Links (ISL).
- The FF simulation must be representative of the interaction between the satellites (e.g. ISL, distributed algorithms, environment and instrument models, operations …). The interactions between the satellites involve Guidance, Navigation and Control functions which allow each spacecraft to be controlled in attitude and position, not only in absolute but also in relative frames. This can be done within;
  - an architecture where each satellite is modelled in an autonomous way (the dynamic and environment simulation is implemented as if it was a single traditional satellite);
  - an architecture based on a cluster simulation. The modelling of all satellite motions is performed using an absolute reference frame to compute the cluster's centre of mass motion, and a local reference frame bound to it to compute each satellite's relative motion. This leads to introduce complex inertia force terms in the relative motion equations. It also leads to the use of an additional set of equations to describe the cluster's centre of mass motion.
- Performance and real time constraints of the simulation must be respected, as for single spacecraft missions. In the case of formation flying though, it implies additional technical challenges related to the parallel simulation of satellites (e.g. time synchronisation, data sharing).

Lifecycle Compatibility

The simulation infrastructure of the Formation Flying testbed must be configured to support the different activities of the system lifecycle. The identified simulator configurations are as follows:

- The **Functional Engineering Simulator** (FES) mainly addresses Phase B. It allows validation of the system concept and functional verification of critical elements of the baseline system design.
- A **Software Validation Facility** (SVF) addresses **Phases C/D.** Its purpose is the validation of on board software in a context representing the spacecraft in its environment. It consists mainly of a simulator of the avionics providing adequate test and debugging tools.
- The testbed for the **Electrical Ground Support Equipment** (EGSE), also addressing Phase C/D, has a two-fold role. First, it gives the capability to simulate in software the whole spacecraft so as to prepare the integration test procedures without involving any hardware. It also supports the AIT/AIV activities of a satellite by simulating the orbital environment and providing a simulation or the actual link to hardware components.
- An **OPerational Simulator** (OPS) addressing Phase C/D-E is used for ground segment integration and validation, validation of flight operation procedures and training of operational staff.

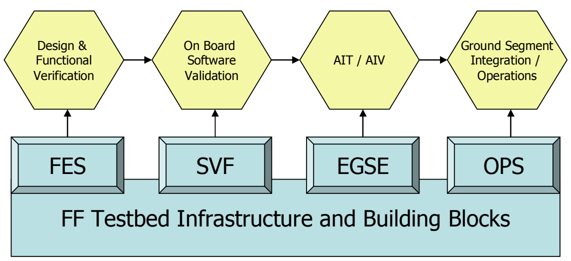

*Figure **1-1**: Simulation Configuration versus System Life Cycle*

The Formation Flying testbed infrastructure ensures the flow of information and time synchronisation between the building blocks of a particular simulator configuration. The different configurations are based on common building blocks, such as:

- models of the equipment, instruments and environment,
- a spacecraft avionics model, common to the SVF, EGSE simulator and operational simulator.

This approach, relying on an evolutive and configurable testbed using common building blocks, offers many advantages. It guarantees a level of homogeneity of the test facilities throughout the project and reduces the development efforts while delivering in the end more precise and mature models. It also gives the possibility to build the system components progressively, in incremental steps. Finally, the testbed supports the design, development, testing and integration of system components (whether software or hardware), even before the final components are available, thus providing good confidence in subsystem specifications, interfaces and performances early in the project.

Model Portability

A model portability approach is used in the Formation Flying testbed. The objective is to allow models to be reused to support different missions and, if necessary, be run within different testbed setups. Two types of models may be considered:

- Generic models than can be adapted to represent specific spacecraft equipment, the orbital environment, the attitude dynamics and the ground segments. Several possible sources exist already (SimVis, VSRF, ESOC).
- Models specific to a particular spacecraft mission that may not be reusable anyway. Spacecraft on board computer simulations fall into the latter category. They are generally spacecraft/platform specific and are closely tied to the onboard processor emulation.

The most widely recognised portability standard is SMP2 which is adopted for the testbed. It is currently partially or fully supported by existing simulation infrastructures such as SIMSAT 4, SimVis BASILES and Eurosim.

Different approaches to adapt existing models to a SMP2 environment are used, such as wrapping of the core code of models (e.g. Matlab/Simulink) within SMP2 components using Mosaic.

Interface with Existing Testbeds

One of the desired FF testbed characteristics is the compatibility with other existing or future test facilities, to be used in a complementary way or even integrated into a unique facility. In this sense, compatibility with other current or future validation facilities is considered. This is namely the case with the ESA VSRF testbed or the PLATFORM testbed from GMV, aimed to demonstrate dynamic aspects of Formation Flying.

2. TESTBED ARCHITECTURE

Overview

The architecture is an open framework connecting simulation nodes, thus allowing the distribution of satellite simulation functions and their parallel execution on the available computing resources (multi core or computer network). The nodes are interconnected by a communication Simulation Backbone.

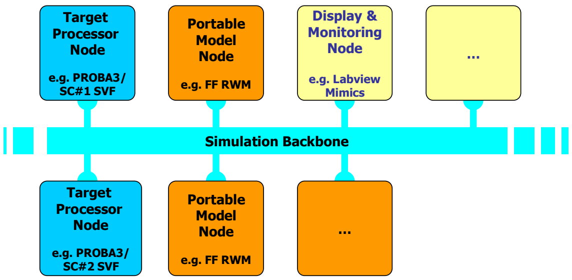

Figure 2-1: Generic FF Testbed Architecture

One can identify the following types of nodes in the architecture:

- *Target Processor Node* (TPN): this node is dedicated to the real on-board software execution (using an instruction set simulator or potentially the real on-board computer) and is the most performance critical node. As a minimum, there is one such node per satellite, dedicated to the execution of the platform on board software. Potentially, there can be additional nodes for a given satellite providing the software execution of the payload or instruments.
- *Portable Model Node* (PMN): this node is foreseen for supporting the simulation of models for which the portability and re-configurability needs are strong and natural. Typically, this type of node can embed the dynamic and environment models as well as the equipment models. The PMN supports fully the SMP2 standard. In the case of FFTB, it is based on the SIMSAT4 framework.
- *Display & Monitoring Node* (DMN): this node is dedicated to aid the user in observing and analysing the simulation from different points of view. A DMN has access to all data available on the Simulation Backbone. From all participating DMN's, some of them may also be responsible for the simulation configuration and commanding of all nodes participating to the simulation.

Possible Configurations

Along the project lifecycle, the above architecture and nodes may be configured while allowing optimal model and building blocks reuse.

FES Configuration

In the example presented below, the FES configuration consists of two satellites simulated on Portable Model Nodes.

The FF testbed infrastructure can be tailored to allow the simulation of a cluster configuration (e.g. 4 nodes) or a centric one (e.g. 3 nodes). It may be used for:

- Robustness analysis allowing to explore the margins of new systems, technologies and algorithms, through extensive simulation.
- Proof of concept and feasibility analyses: the system is modelled and its main characteristics are analysed on case-by-case simulations.
- Identification of critical technologies, initiation of equipment and function requirements and specifications.

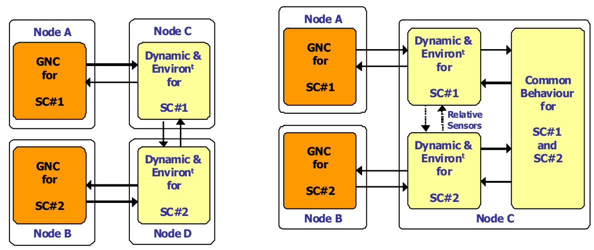

Figure 2-2: FES Architecture Configuration

SVF Configuration

The role of the SVF is to execute real flight software on a simulated on board computer and to provide all facilities to analyse and debug this software.

For most standalone OBSW validation activities, there is no need to simulate the other spacecraft belonging to the formation. It is not even required to have the environment models in closed loop for all tests. The first SVF configuration focuses thus on the standalone OBSW validation obtained with a single target processor.

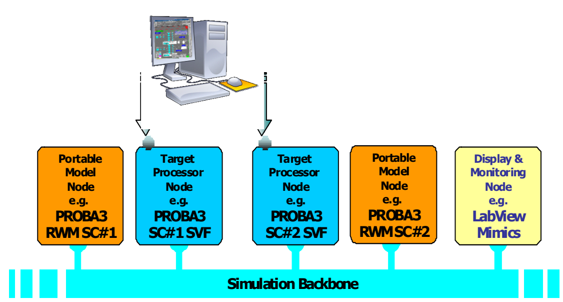

Figure 2-3: Complete SVF Architecture

System level tests involving FF and GNC algorithms require the simulation of the environment, equipment and of the other satellites of the formation flight. The corresponding configuration may be obtained from the FES configuration where the "GNC, MVM, FDIR" node is replaced by the SVF as shown in the figure.

OPS/EGSE Configuration

In the EGSE configuration, the simulator is connected to the EMCS (SCOS) through an EGSE router. This connection is done at the level of the TPN and does not involve the simulation backbone. The EGSE configuration may also integrate hardware in the loop. In that case, the FF testbed can be interfaced to real (or breadboard) Equipment Under Test (EUT).

The EUT has a hardware interface, e.g. RS232, MIL-BUS-1553, SpaceWire etc, and a communication protocol is used to perform the information exchange between control software running either in the TPN or in a breadboard computer. Therefore, the FF testbed is required to support interfaces between the EUT and the other nodes participating to the simulation (simulated OBC, the environment and dynamics models, the simulated hardware …)

In the illustrated configuration, the TPN has to be representative of the real TM traffic that would be expected from the spacecraft through the ground station and the BBE. To achieve this, a BBE simulation can be introduced in the TPN and interfaced with the EGSE router.

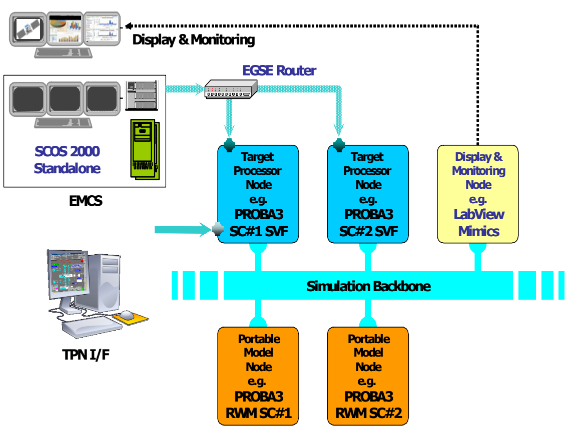

Figure 2-4: OPS/EGSE Architecture Configuration

3. TESTBED DEMONSTRATION CASE

Baseline

The FF testbed is demonstrated with an application based on the PROBA-3 mission concept. The demonstration application covers the various FF testbed configurations and the FF specific aspects, as well as the possibility to easily evolve from one configuration to the other.

The following baseline is the starting point to build the demonstration architecture and cases.

- The spacecraft avionics model will be reused from PROBA-2 (with PROBA-2 SVF adapted as TPN and reused ADPMS/ OBSW).
- The GNC applications, models, and manoeuvre scenarios are reused from previous FF studies and demonstrations.
- The FF MVM and FDIR is implemented as a state machine able to generate representative data. Scisys MMOPS from Aurora will be reused to demonstrate usability for mission planning.

PROBA-3 Demonstration Scenarios

A complete set of scenarios aimed to demonstrate the suitability of the FFTB for the PROBA-3 mission has been identified in the table below. The scenarios cover different mission phases involving functional models of GNC, FFM, MVM and FDIR. During the validation of the PROBA-3 FF system, it will be straightforward to substitute the FFTB functional models with the PROBA-3 ones.

Scenario	Description
Coarse Formation Acquisition	Starting from 10 km relative distance the formation is acquired up to a relative distance of 150 m +/- 10 m.
Fine Formation Acquisition and Station Keeping	Starting from 150m +/- 10m relative distance the Fine Formation is acquired and maintained.
Perigee Pass	Starting from Fine Formation, the perigee pass manoeuvre is computed and executed.
Formation Resizing	Starting from Fine Formation at 150 m, the formation is resized to 25 m.
Formation Retargeting	Starting from Fine Formation at 150 m, the formation is retargeted 30º.
Control via ISL	Starting from Fine Formation at 150 m the coronograph (A) sends control via ISL to the occulter (B) to perform a resizing manoeuvre to 250 m.
Metrology via ISL	Starting from Fine Formation at 150 m. The coronograph (A) sends to the occulter (B) the metrology measurements, the occulter (B) computes and performs a formation resize to 25 m
Metrology and Control via ISL	Starting from Fine Formation at 150 m.. The coronograph (A) sends to the occulter (B) the metrology measurements, the occulter (B) computes the station keeping manoeuvre and sends the control via ISL to the Coronagraph(A).
Collision Avoidance	During a formation resizing manoeuvre a thruster failure injects the spacecrafts in collision orbit.

4. CONCLUSION

The FFTB infrastructure addresses the main challenges of formation flight missions testing in term of re-use and re-configurability of the models, isolation and parallel execution of several simulation nodes.

It allows the overall reconfiguration of the system in function of the evolution of the models and the availability of hardware components. It allows model migration or evolution, during the concurrent development, especially when switching from study models to executable on board software. Finally it allows reuse of test procedures across the life cycle.