Space Flight Technology, German Space Operations Center (GSOC)
Deutsches Zentrum für Luft- und Raumfahrt (DLR) e.V.

Formation Flying Testbed

J.S. Ardaens, S. D’Amico

Doc. No.	:	TN 09-01
Version	:	0.2
Date	:	26.02.2009

Document Change Record

Issue	Date	Pages	Description of Change
0.1	Feb. 18, 2009	all	Document structure
0.2	Feb. 26, 2009	all	Finalized system description and overview

Table of Contents

Scope

This document describes the GNSS-based Formation Flying testbed used at GSOC.

Acronyms and Abbreviations

ASCII	American Standard Code for Information Interchange
FCC	Flight Control Computer
FFT	Formation Flying Testbed
GCC	GNU Compiler Collection
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSOC	German Space Operations Center
OBC	Onboard Computer
PPS	Pulse per Second
RTEMS	Real-Time Executive for Multiprocessor Systems
SPARC	Scalable Processor ARChitecture
SW	Software
TC	Telecommand
TCP	Transmission Control Protocol
TM	Telemetry

1. Motivations

The use of GNSS sensors is widely spread in the design of spacecraft orbiting below the GNSS constellations. They provide accurate 3D position information in real-time and are thus natural candidates to serve the needs of spacecraft guidance, navigation and control (GNC). In addition the GNSS observables can be processed post-facto for very precise orbit restitution. However the utilization of GNSS sensors is not trivial and their behavior might be difficult to model, which justifies the need of hardware-in-the-loop test facilities.

A GNSS receiver processes only radiofrequency waves sent by a constellation of satellites to derive the position information. As a consequence it is possible to recreate the physical environment measured by the sensor with a dedicated device, called GNSS signal simulator. The use of a GNSS signal simulator is of great interest for the design of test facilities with a high degree of realism. It becomes possible to use the true hardware sensor in flight conditions and to connect it directly to the computer on which the flight software is running. The flight software implements the sophisticated GNC algorithms to be tested whose control actions are fed back to the simulation environment.

A real-time hardware-in-the-loop, closed-loop testbed is of relevance for advanced functional and performance test. The use of the real receiver in the loop ensures that the flight software is able to interface properly the sensor and to handle the failures that may occur. Ultimate assessment of navigation, guidance and control performances is achieved by avoiding any software model of the sensor. Furthermore the introduction of a target computer representative of the onboard computer on which the flight software is running may serve as support for software resource usage analysis.

The Formation Flying Testbed (FFT) developed at GSOC is intended to support the design, implementation, testing and validation of real-time embedded GPS-based GNC systems. It allows the definition of complex scenarios involving up to 2 spacecraft and equipped with L1-GPS receivers.

2. System Architecture

2.1 Design Drivers

The Formation Flying Testbed is a mission-independent test facility. As such it is designed to offer a high flexibility in order to be utilizable for a large variety of mission profiles. Focus has been given to the simplicity and transparency of utilization to ensure the durability of the project. To that end the complete simulation environment is implemented using Simulink, which offers a user-friendly interface, a high modularity and is usually well-known among the aerospace community. Furthermore it makes the complete system easily upgradeable and extendable. Dedicated sophisticated user-defined Simulink blocks have been developed using C++ based S-functions to support the steering of the system and the communication between components.

2.2 System Overview

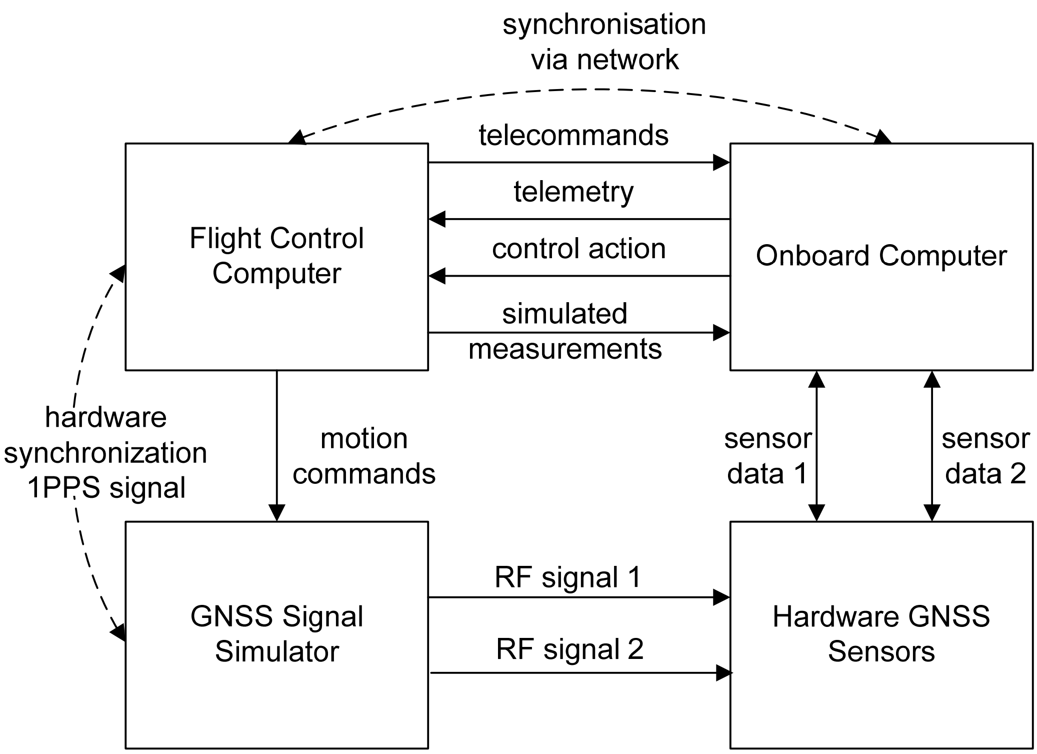

Fig. 1: Overview of the Formation Flying Testbed

The system comprises four logical units (cf. Fig. 1):

- **Flight Control Computer (FCC)**: model-based Simulink application that simulates the spacecraft dynamics and characteristics. It computes the transational and rotational motion of the spacecraft using precise force modeling and including the control actions sent by the GNC flight software. Furthermore it comprises accurate models of the spacecraft actuators and additional sensors. The FCC steers the GNSS signal simulator by providing motion commands in real-time. Complex scenarios are built using time-dependent events. A simplified emulation of the ground segment is also available to send telecommands and receive telemetry data.

- **GNSS Signal Simulator (GSS)**: device used to simulate the radiofrequency signals output of the GNSS antenna of the spacecraft. It takes as input the position, velocity, and attitude of the spacecraft sent by the FCC. Additional parameters can be tuned to improve the realism of the scenario: description of the GPS constellation, atmosphere model, antenna characteristics.
- **Hardware GNSS Sensors**: receivers used to process the radiofrequency signals out of the GSS and feeding the onboard computer
- **Onboard Computer (OBC)**: device on which the GNC flight software is running. The control actions are sent in real-time to the FCC. Depending on the maturity of the GNC prototype to be tested, the OBC can be for example a x86 architecture running a Simulink model or a SPARC architecture running C/C++ flight software compiled for a real-time operating system.

2.3 System Timing and Synchronization Requirements

The FCC runs and sends commands to the GSS at a fixed frequency which can be either 1Hz or 10Hz. In the current implementation the FCC triggers the tasks executed by the flight software at the same rate.

The time synchronicity between all the testbed components is a mandatory requirement for a real-time test facility.

Requirement : The logical units shall be synchronized within the simulation sample time.

Rationale : Since the motion commands are precisely time-tagged it is just necessary that the motion commands arrive to the GSS a fraction of second prior to their time of applicability. The GNSS sensors are precisely synchronized with the GSS and output time tagged messages that are processed afterwards by the GNC algorithms. The resulting control action is included in the spacecraft dynamics with some delay. In order to avoid unexpected behavior this delay shall not exceed one simulation step.

2.4 Time synchronization

Because of different qualities of the hardware clocks (whose drift can reach several seconds per day) it is not possible to let each component to rely on its own clock. The GNSS Signal Simulator owns a very precise and stable clock and is consequently considered as reference clock for the testbed. The requirement on synchronicity just ensures that no long-term clock drift occurs between the components but is not too stringent regarding the synchronization accuracy. As a consequence there is no need for a rigorous real-time operating system. The FCC is synchronized with the GSS with a hardware dedicated interface. The GSS emits a 1 pulse per second (1 PPS) electrical signal which is then used for clock synchronization at the millisecond level by a timer card installed on the FCC.

The synchronization between the FCC and the flight computer is done by sending every second a synchronization message using a networking protocol. This solution does not guarantee rigorous time synchronization but achieves synchronization performance below 10ms. Depending on the network traffic and the load of the operating system this synchronization is fully acceptable for the practical needs of the testbed.

2.5 Communication

The transmission of data between the FCC and the GSS/OBC is done using the TCP protocol. Tanks to its universality, simplicity of implementation and robustness, this choice appears to be the most suitable for the testbed architecture. Note that rigorous real-time operations are not guaranteed and a system sample time of less than 100ms is not recommended.

3. System description

3.1 GNSS Signal Simulator

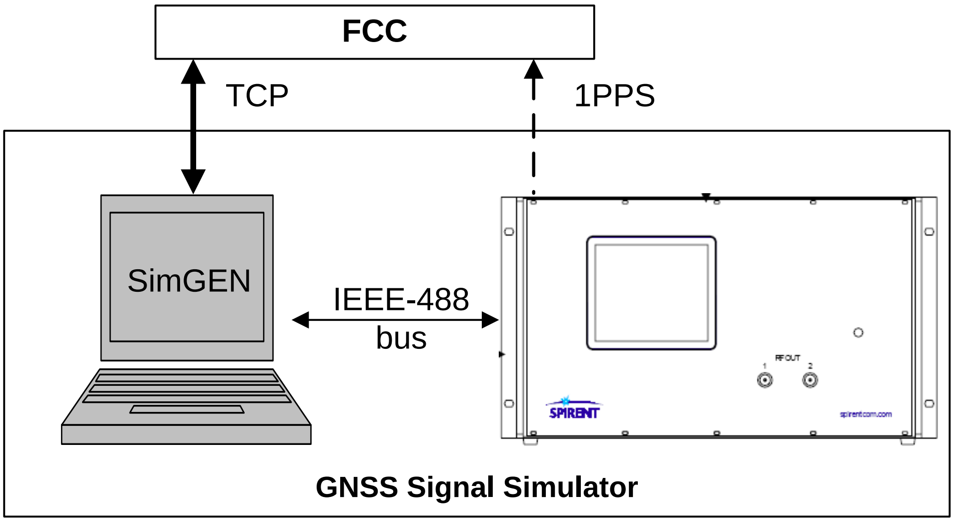

Fig. 2: Graphical overview of the GNSS Signal Simularor

The GNSS Signal Simulator system comprises two components. The hardware part is a Spirent GSS7700 device [2] able to provide RF signals for two independent antennas on the L1 transmitting frequency. As a consequence it is possible to define scenarios comprising two spacecraft with one L1-antenna or only one spacecraft with two antennas. The device is steered by a simulator PC Software, called SimGEN, defining the scenario used to generate the RF signals. SimGEN triggers the GSS7700 using an IEEE-488 interface that guarantees communication capabilities without any delay. In addition SimGEN owns a remote access component which allows steering by a remote system [3]. This functionality is used by the FCC to provide motion commands and other scenario parameters to SimGEN using a TCP interface. The GSS7000 provides as output a 1PPS signal which is used to synchronize the FCC.

3.2 Flight Control Computer

The FCC is a standalone PC equipped with programmable digital I/O and timer board for the synchronization with the 1 PPS signal coming from the GSS. Table 1 summarizes the characteristics of the FCC.

Table 1: FCC technical characteristics

Processor	Intel Pentium 4 2.4 GHz
RAM	500MB
Hard drive	200GB
Digital I/O and timer board	Amplicon PC215E
Operating system	Windows XP

The software part of the FCC is implemented using a Simulink model running in a Microsoft Windows environment. The software architecture is described in Fig. 3. The Simulink model runs at a fixed-size sample time which can be either 100 ms or 1s and uses a 5^{th} order Dormand-Prince method for the numerical integration. In order to synchronize the Simulink steps with the sample time of the testbed the *SimGEN interface / hardware time synchronization* block is implemented as a blocking thread that returns only if the time elapsed since its last call exceeds the model sample time.

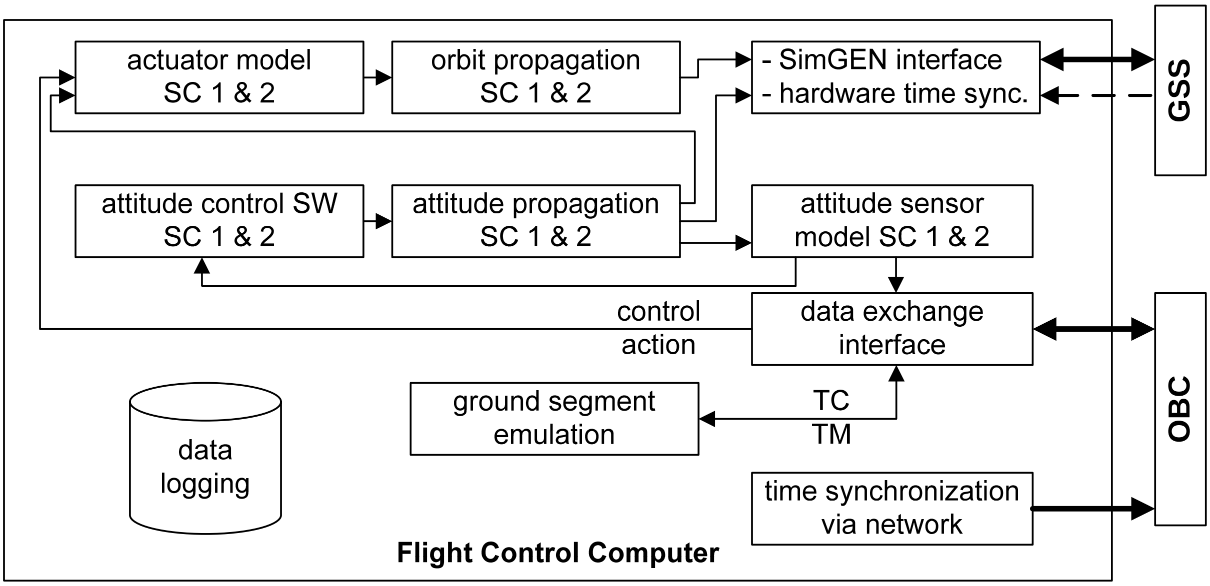

Fig. 3: Software architecture of the Flight Control Computer

Currently the FCC can handle up to 2 spacecraft (SC) and comprises the following functionalities:

- **Actuator model**: applies random execution errors to the time-tagged thrust commands inputting the orbit propagator.
- **Orbit propagation**: propagates the state of the spacecraft at the current time of the simulation using up-to-date force models provided in the GHOST C++ library. The block makes use of the integration methods provided by Simulink and thus depends on the simulation step. The following forces are included:
    - gravitational force using an external gravity model file
    - luni-solar perturbations using analytical ephemerides for the Sun and Moon
    - drag modeling using an external flux data file and the Gill-Jacchia atmospheric density model
    - tidal effects
    - relativistic effects
    - solar radiation pressure using a simple cannon ball model

  The block takes as input a time-tagged thrust to be applied to the spacecraft. The start and end time of the maneuvers are quantized to the Simulink simulation step which induces artificial maneuver execution errors. Depending on the burn duration this force will be considered as impulsive maneuver or extended maneuver to limit the modeling errors due to the inclusion of maneuvers.
- **SimGEN interface / hardware time synchronization**: takes as input the position, velocity, acceleration, jerk, attitude, angular velocity, angular acceleration, angular

jerk of up to two spacecraft at the current simulation time and transmit the corresponding motion messages to the GSS [3].

- **Attitude control SW, attitude propagation, attitude sensor model**: not implemented yet. Currently an attitude profile is simply defined through a sinusoidal error pattern and some random noise
- **Ground segment emulation**: set of Matlab scripts defining time-tagged telecommands to be sent to the onboard flight software. Not fully implemented yet.
- **Data exchange interface**: sends/receives data using the TCP protocol to the flight software. Thanks to a modular approach, any kind of data can be sent/received: attitude information, maneuver commands, GPS receiver commands, TC/TM, etc…
- **Time synchronization via network**: sends at each simulation step a synchronization message to trigger to execution of the flight software.
- **Data logging**: logs relevant data generated by the FCC and received from the OBC.

3.3 Hardware GNSS Sensors

Currently the testbed uses single frequency Phoenix GPS receivers [4]. The signals coming from the GSS are amplified by low-noise amplifiers and fed to the two receivers. The receivers provide user-selectable Mitel messages sent in ASCII format using serial ports to the flight software. In particular the messages can be kinematic navigation solutions and/or raw measurements which are then processed by sophisticated navigation filters running on the OBC.

3.4 Onboard Computer

The onboard computer is the target unit on which the flight software is running. Generic modules and interfaces are developed to support the rapid integration of the GNC algorithms into the testbed (cf. Fig. 4).

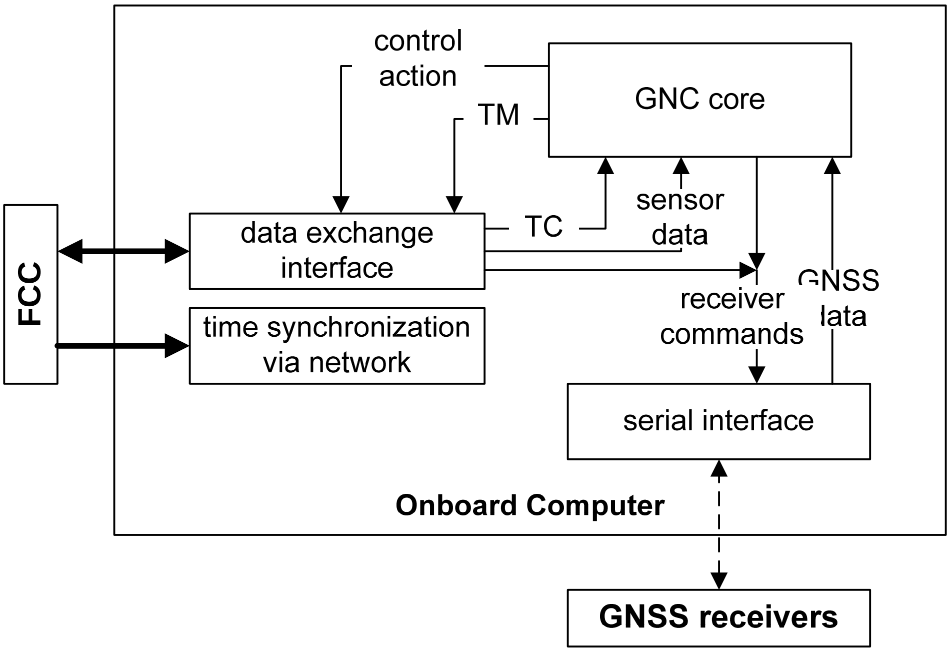

Fig. 4: Architecture of the flight software

The onboard computer comprises the following items:

- **GNC core**: mission dependent GNC algorithms to be tested and validated.
- **Data exchange interface**: sends/receives data using the TCP protocol to the FCC. Thanks to a modular approach, any kind of data can be sent/received: attitude information, maneuver commands, GPS receiver commands, TC/TM, etc…
- **Time synchronization via network**: blocking thread waiting for a synchronization message coming from the FCC. The thread returns once the synchronization message is received. This synchronizes the flight software execution steps with the FCC simulation steps.
- **Serial interface**: supports the communication between the flight software and the GNSS receivers

Except the GNC core, all the components are generic items that are available for every project and are easily configurable.

The target onboard computers can be of several types. Currently the following systems are supported:

Table 2 : Supported hardware for the OBC

Architecture	Operating System	Development environment	Status
x86	Windows XP	Simulink	mature
LEON3 board	RTEMS	GCC	under development

3.5 Specifications of the GNC core

- Only an implementation using a Simulink model is currently supported.
- The model must use a fixed-size simulation step.
- No constraints are set for the interfaces. A short configuration of the testbed is however required to integrate properly the GNC core into the OBC.
- The GNC core can be implemented using either native Simulink functions or S-functions which allow the usage of any C/C++ or Fortran code.

Conclusion and way forward

A real-time, hardware-in-the-loop, GNSS-based formation flying testbed is now available to support the GNC activities at GSOC. The FFT has achieved now a mature status and can be used to test real-time GNC algorithms.

The PRISMA flight software has been successfully tested using this environment (NB: a small test report will be added to the technical note). This closed-loop test using real hardware GPS receivers demonstrates the proper functioning of the interfaces and the overall consistency of the simulation environment.

Many improvements and tests are still necessary to enhance the realism of the testbed and its ergonomics. A non-exhaustive list of desired features is provided hereunder to drive the further developments:

- Support of the LEON3 board as onboard computer.
- Time Synchronization via network using NTP protocol.
- Development of an attitude propagation module for the FFC.
- Emulation of attitude control software for the FCC.
- Development of a generic data logging module.
- Development of a remote and user-friendly emulation of the ground station.

References

[1] Spirent Communication, “SimGEN Software User Manual “, Issue 1-22, DGP00686AAA,2007.

[2] Spirent Communication, “Signal Generator Hardware User Manua”l, Issue 1-13, DGP00703AAA, 2007.

[3] Spirent Communication, “SimREMOTE User Manual”, Issue 2-02, DGP00792AAA,2007.

[4] Montenbruck, O., and Markgraf, M., “User’s Manual for the Phoenix GPS Receiver”, DLR/GSOC; V1.6, 2005.