



NLR-TP-2005-420

## **Precision Agile Control Systems: Developments in Test and Verification Equipment for Spacecraft**


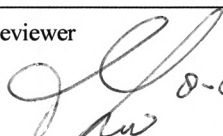

B.A. Oving and T. Zwartbol

This report has been based on a paper presented at DASIA2005, at Edinburgh (UK) on 01-06-2005.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Customer: National Aerospace Laboratory NLR  
Working Plan number: AS.1.I, AS.1.A, AS.1.D  
Owner: National Aerospace Laboratory NLR  
Division: Aerospace Systems & Applications  
Distribution: Unlimited  
Classification title: Unclassified  
July 2005

Approved by:

Author  1-8-05	Reviewer  8-8-05	Managing department 
---	---	--



## Summary

This paper presents the application of Control Momentum Gyros (CMG) for agile pointing of spacecraft, development of On-Board Software (OBSW) to control the CMGs, and the accompanying Real-Time Test Bench (RTTB) for the Precision Agile Control Systems (PACS) study from ESA. The PACS study is conducted by Alcatel Space Industries (France) as prime contractor with subcontractors DEIMOS (Spain) for the development of on-board software and the National Aerospace Laboratory, NLR (the Netherlands) for the development of the RTTB.

The study encompasses the development and real-time testing of the on-board software (OBSW) for a precise and agile spacecraft attitude control and measurement system (ACMS), featuring a cluster of four Control Moment Gyros (CMGs) and the development of the RTTB for this purpose. The objective of the study is to perform real-time closed-loop performance tests of a global guidance algorithm for fine pointing of an agile ACMS. For that purpose, a dynamic real-time test bench should be developed using to a maximum extent auto-coded S/W components from Matlab/Simulink sources, for both on-board software and real-time simulation.

The PACS project constitutes a major milestone for the ACMS development of the future High-Resolution Earth Observation Satellites that require both a high agility of the line-of-sight and a very fine pointing and stability. It offers the opportunity to implement, test and validate a complete ACMS based on CMGs.



## **Contents**

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Development of Attitude Control Loop and Simulator</b>	<b>7</b>
<b>3</b>	<b>Development of On-Board Software</b>	<b>9</b>
<b>4</b>	<b>Development of Real-Time Test Bench</b>	<b>11</b>
<b>5</b>	<b>Conclusion</b>	<b>16</b>
<b>6</b>	<b>Acknowledgements</b>	<b>17</b>
<b>7</b>	<b>References</b>	<b>17</b>

## 1 Introduction

The case study for PACS project is the SPECTRA mission, which was a candidate mission in the framework of ESA's Earth Explorer mission. SPECTRA stands for *Surface Processes and Ecosystem Changes Through Response Analysis* and aims to study the interaction between vegetation and climate (CO<sub>2</sub> carbon cycle).

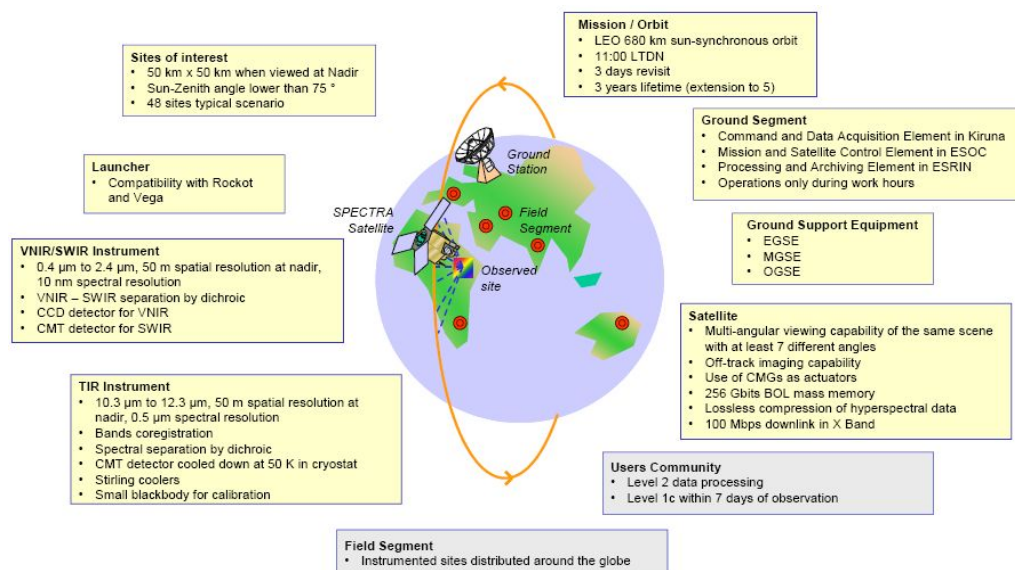
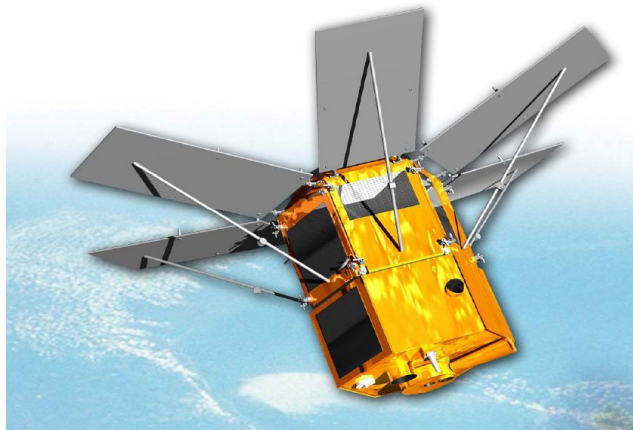


Fig. 1 Overview of SPECTRA mission

The objective of SPECTRA requires the acquisition of 7 images of the same ground target under 7 different angles. This requires precise and swift attitude manoeuvres, to be realised by Control Moment Gyros (CMG) that can produce large moments around spacecraft axes. Thus, SPECTRA is a mission that can improve its performance by the use of an efficient attitude control with the help of CMG's.

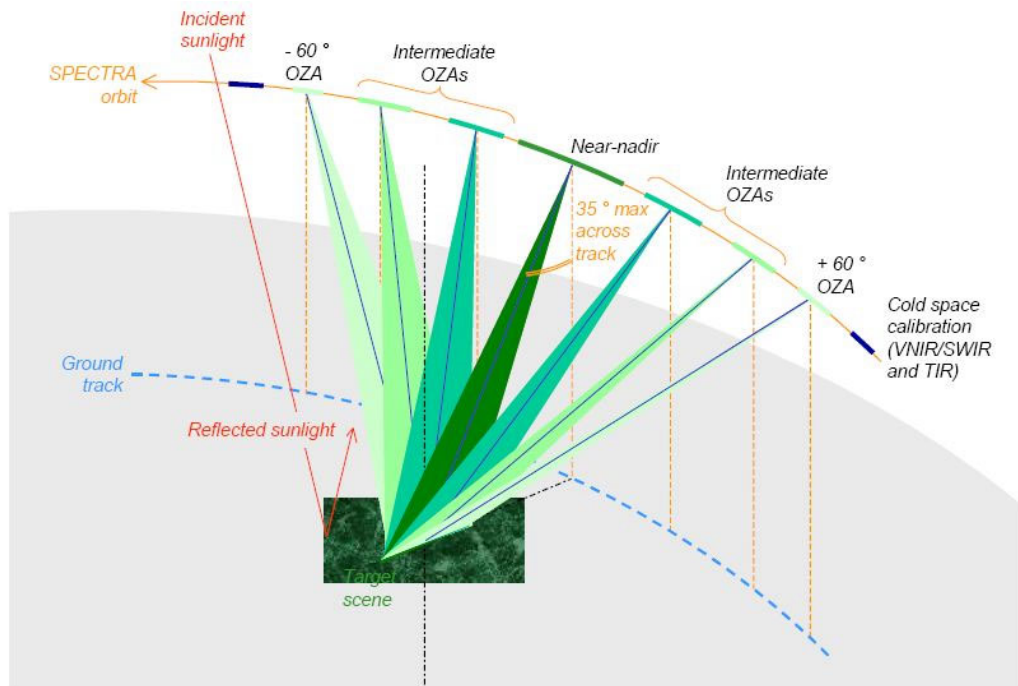


Fig. 2 SPECTRA image acquisition

In the PACS project, the ACMS is based on four CMGs in a pyramidal configuration, which fits well within the SPECTRA mission. The singularity avoidance law is especially efficient when the satellite manoeuvre remains almost constant in direction, which is the case in the Bi-directional Reflectance Distribution Function (BRDF): a succession of 7 image acquisition phases.

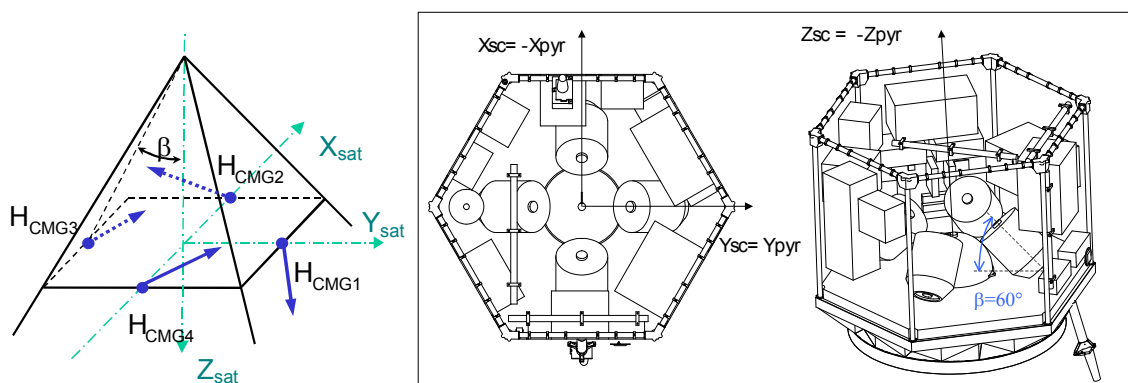


Fig. 3 CMG pyramidal layout

There are some constraints for the layout of the CMGs. First, the pyramid angle  $\beta$  is determined by the requirement that attitude control shall still be possible with a failure of one CMG. This

leads to a pyramid angle to be  $60^\circ$ . Second, there is a mechanical constraint limiting the orientation, as the satellite structure must accommodate the CMGs.

The main objective of the PACS project is to verify the capability of the CMGs as actuators in an ACMS in a Real-Time Test Bench (RTTB). Software simulation had already shown their applicability, but the PACS RTTB uses a real breadboard CMG mounted on a three-axes rotation table (T3X) in a Hardware-in-the-Loop (HIL) simulation.

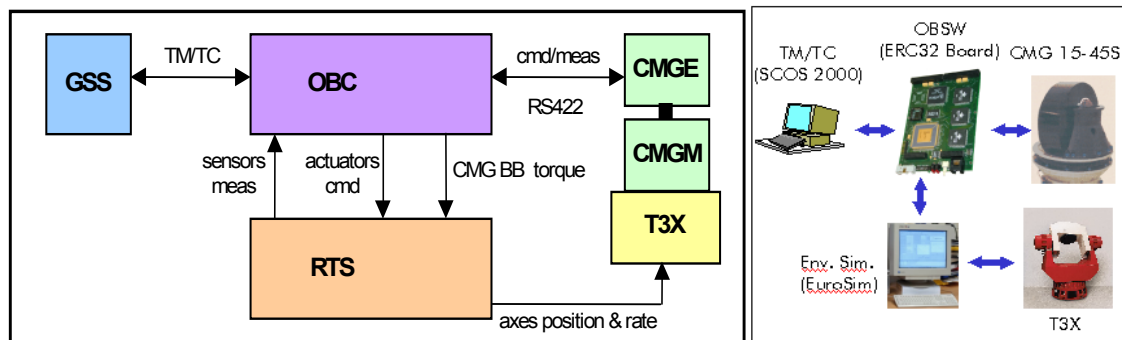


Fig. 4 RTTB set-up

The Ground Station Simulator (GSS), or TM/TC client, is used for spacecraft command and monitoring (TM/TC). The On-Board Computer (OBC) emulator is based on the target hardware (ERC32 processor) and runs the On-Board Software (OBSW). The Real-Time Simulator (RTS) is based on EuroSim and contains simulation models, previously created in Simulink/Stateflow and converted to C-code with the tool Real Time Workshop (RTW). The target CMG Electronics (CMGE) and CMG Mechanics (CMGM) are to be put on top of a 3-axes rotation table (T3X), which is commanded by the RTS.

## 2 Development of Attitude Control Loop and Simulator

Alcatel Space have gained experience with CMG equipment and application thereof early in the definition phases. The CMG gimbals control law, the CMG equipment, and the ACMS are closely coupled. The nominal ACMS attitude control loop provides gimbals commanded position and rate profiles at ACMS sample frequency (16 Hz) for each CMG, whereas the CMG gimbals control loops themselves operate at a frequency of 96Hz.

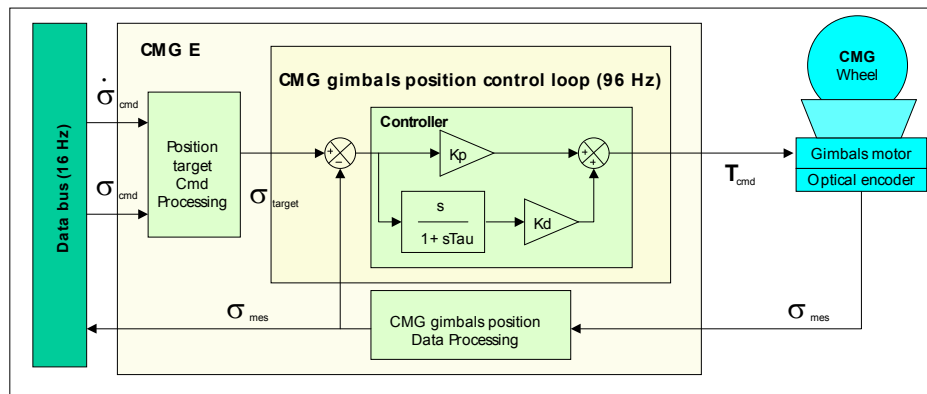


Fig. 5 ACMS control loop (16Hz) and CMG control loop (96 Hz)

Alcatel Space have developed a high-performance ACMS with an efficient spacecraft mode logic and a CMG guidance law adapted to each spacecraft mode, from separation from the launcher until the observation mode.

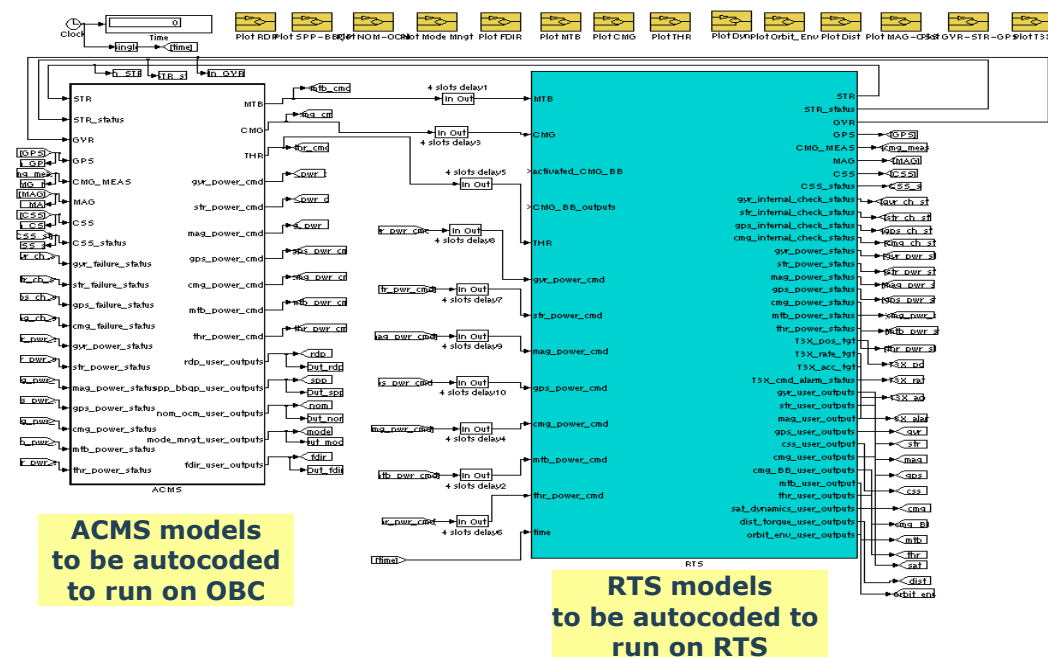


Fig. 6 Simulator development in Matlab/Simulink/Stateflow

An important aspect of the PACS project is the application of autocoding, for both the generation of the real-time simulation software as well as the on-board application software. For the early design and development of the ACMS on-board application software, Alcatel Space have developed an all-software simulation in Simulink/Stateflow, for both the dynamics simulation of sensors and spacecraft and actuator dynamics, as well as the ACMS on-board application software.

For the verification and validation of the ACMS concept via real-time HIL tests in the RTTB, the Simulink/Stateflow model of the spacecraft sensors and spacecraft and actuators dynamics, disturbances and environment was converted into C-code using the Real Time Workshop (RTW) tool of the Matlab tool set. The C-code, thus generated, was directly used as real-time Simulation Model Software (SMS) in the EuroSim simulation environment of the RTTB. C-code generated by RTW can be easily to the EuroSim simulation environment using the Mosaic tool of NLR.

One of the study objectives was to use code generation for the generation of the on-board application software, as well. So, the Simulink/Stateflow model of the ACMS on-board application software was converted into C-code with RTW. Again, the code thus generated was directly used in the On-Board Software (OBSW) running in the OBC emulator.

In principle, the employed autocoding paradigm ensures a rapid prototyping of software with inherent validation of the code. Still, some effort has been put into the validation of the real-time simulation (open-loop testing). Simulator validation uses 7 elementary closed-loop tests of various transitions between spacecraft modes (LEOP modes, operational modes, safe hold modes).



### 3 Development of On-Board Software

The OBSW for the ACMS is targeted for the ERC32 processor and can be divided into four main architectural components:

- Basic Software, or OBBSW, which include the I/O device drivers (watchdog, RS422 serial lines, VME bus), the implementation of certain Packet Utilisation Standard (PUS) services (Telecommand Verification, Housekeeping, Event Logging & Reporting, Function Management) and their associated Telecommand and Telemetry source packets, and the protocol for exchanging data with the RTS.
- ACMS Application Software, or OBASW, consisting of cores implementing Modes Management, System-level FDIR policy, Autonomous functions and Control Laws for the CMGs.
- OBASW Shell, which interfaces the ACMS Application Software with the Basic Software.
- A background task that measures the CPU load over each 16Hz cycle. The average considered over every 8-second period is reported in the Housekeeping telemetry.

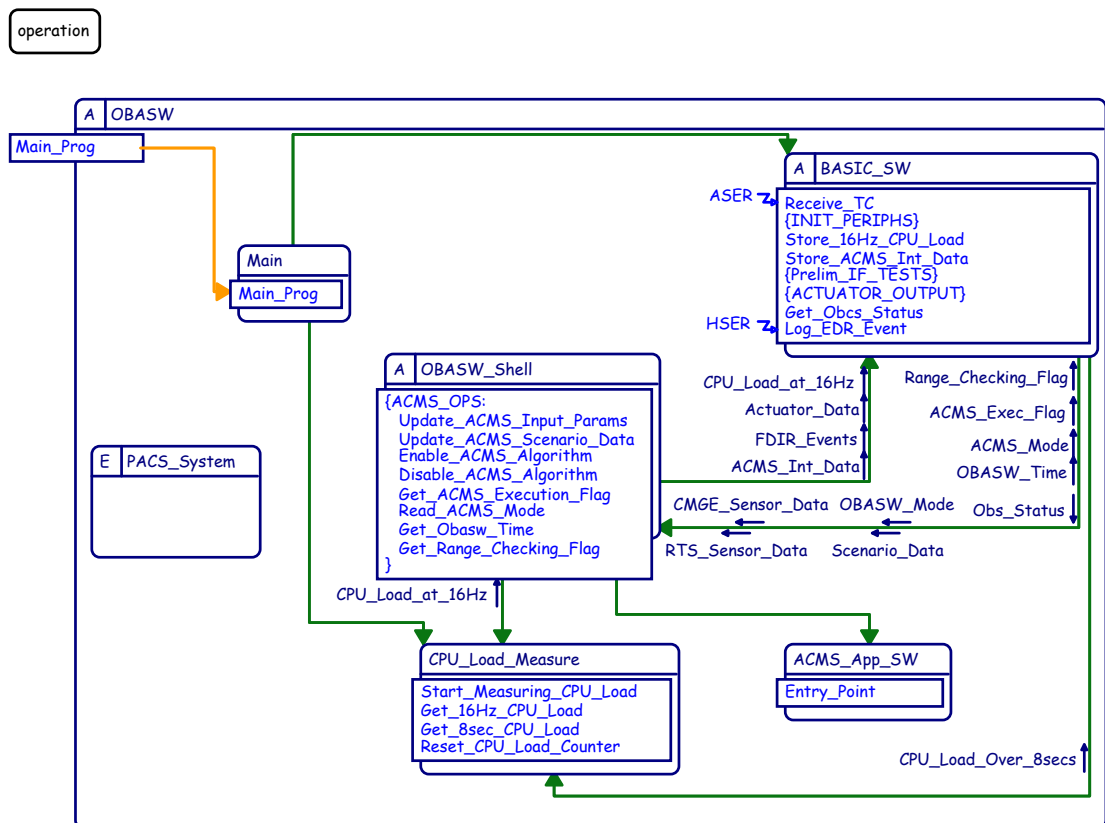


Fig. 7 OBSW architecture

The OBBSW interfaces with the CMGE via RS232, with the GSS (TM/TC) via RS232, and with the RTS (EuroSim) via VME and the reflective memory ScramNet interface.

The software development follows the standard waterfall life cycle, using HOORA/UML with Use Cases and Sequence Diagrams for requirements definition, supported by “Opentool” from TNI. The design is based on the HRT-HOOD method, supported by the “Stood” tool from TNI.

Implementation of the ACMS Application Software is done using autocode generation from the Matlab/Simulink models, resulting in ANSI C code. The OBBSW and OBASW shell are programmed in Ada95, constrained by Ravenscar profile. The operating system for the ERC32 is the AONIX ObjectAda RT RAVEN compilation chain and RTOS.

Several tests have been performed to verify and validate the OBSW:

- Software unit and integration tests using TSIM/ERC32 and GDB/DDO, both tools running on a Linux PC.
- Open-loop validation tests on a Tharsys ERC32 VME On-Board Computer (OBC emulator).
- TM/TC interface tests exchanging data between ERC32 and GSS on a Linux PC via RS232.
- CMG interface tests exchanging data between ERC32 and CMG Interface Simulator on a Linux PC via RS232.
- RTS interface tests synchronising and exchanging data between ERC32 and RTS via ScramNet reflective memory / VME bus.

The OBSW development within the PACS project encountered some challenges, specifically on the low-level interaction with the hardware. Some lessons learned:

- ✓ Ada/C interface implementation and RTW autocoding was easy.
- ! To avoid an VME Instruction Access Error when VME bus is accessed, it was found out that one wait state on RAM had to be set. This was a peculiarity of the OBC emulator board produced by Tharsys.
- ! An exchange of data with CMG electronics must be completed within a 96Hz cycle ( $\approx 10.4\text{ms}$ ). Consequently, the serial line must operate at 115200 Baud. However, the Tharsys OBC emulator board did not have serial port hardware buffering, so software interrupt servicing overhead was very high (and there was less than  $87\mu\text{s}$  between interrupts). This has lead to a design where the OBASW had to avoid parallel interrupts (e.g. from TM/TC side), which imposed a strong design constraint.
- ✓ Integration with AONIX RAVEN RTOS on ERC32 was simple (although, improved support for Ada exception handling would be appreciated).
- ✓ ACMS execution frequency achieved.
- ✓ Overall RT-TB synchronisation achieved.

## 4 Development of Real-Time Test Bench

At the beginning of the project, a requirement change was introduced: the required closed-loop frequency of the RTTB was increased from 16Hz to 96Hz (10.417 msec interval). A feasibility study in phase 1 of the project showed that the hardware available at that time (Linux PC and SBS PCI-VME interface) caused too much data exchange timing jitter on the interface between RTS and VME FE. A decision was made to switch to a ScramNet reflective memory interface and a SGI Octane dual-processor running the real-time IRIX 6.5 OS.

In the second phase of the project, it was decided to slave the RTS to the On-board Computer (OBC) using interrupts at 96Hz. Implementation and tests were performed to show the feasibility. Furthermore, ANSI C code was generated by RTW from the Matlab simulation and exported to EuroSim using the automated conversion features of the NLR Mosaic tool.

A dedicated schedule was created to separate the 96Hz simulation & OBC interface from the 8Hz T3X interface. Extensive verification and validation tests have been performed and, after a training session, the RTTB was accepted.

Phase 3 was the concluding phase of the project in which the accepted RTTB was used to conduct closed-loop tests with not only the OBC as hardware-in-the-loop, but also the CMG breadboard on the rotation table.

The following schematic represents the architecture of the RTTB, showing its most important components.

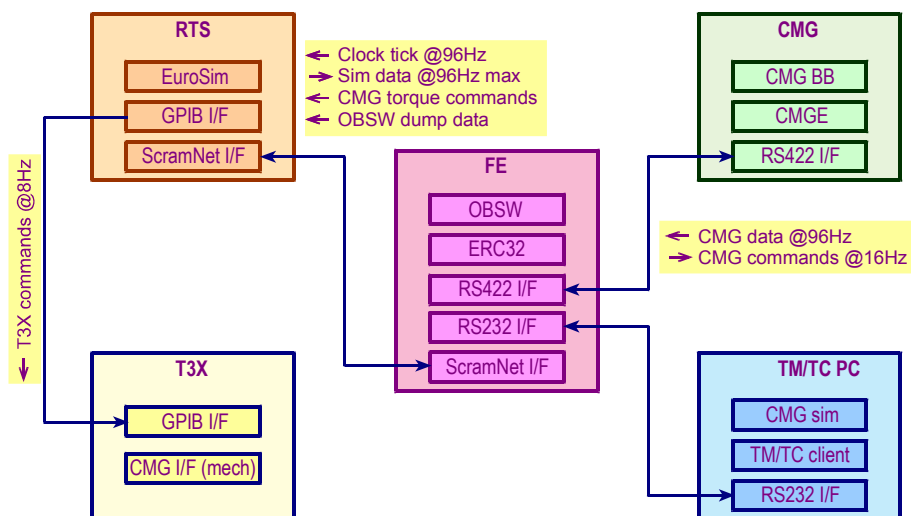


Fig. 8 RTTB architecture

The main component to be developed for the RTTB is the RTS, incorporating:

- Simulation Model Software (SMS), containing models of the spacecraft sensors & actuators (CMGs), dynamics, disturbances, and environment;
- Interface Software, implementing the interfaces to OBC and T3X.

The SMS is exported from Matlab/Simulink into the real-time simulation environment EuroSim using Real-Time Workshop and Mosaic tool of NLR. The RTS has two interfaces: a real-time, high-speed, high-data-rate ScramNet interface to the FE and a real-time, low-data-rate GPIB interface to the T3X. The interface software is developed in C and integrated into EuroSim. The other RTTB components are either Customer Furnished Items (CMG, T3X) or have already been developed in this project (OBSW, CMG sim, GSS TM/TC client).

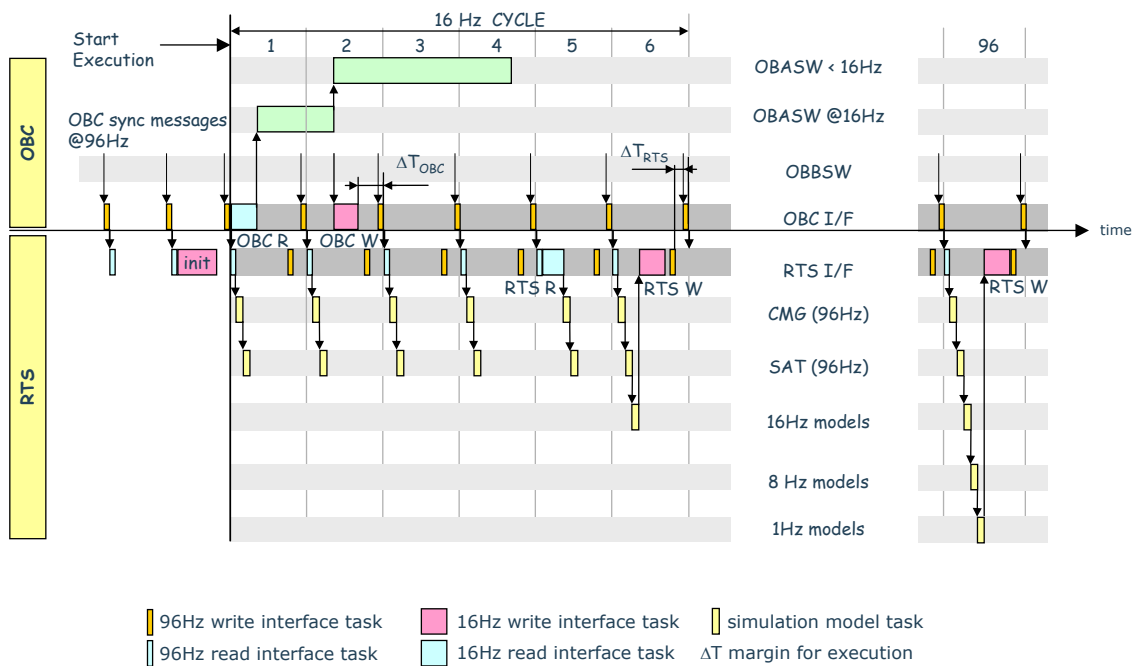


Fig. 9 RTTB chronogram

The 96Hz closed-loop interaction between RTS and OBC is the main challenge for the interface software, considering the need for synchronisation of both components. The chronogram shows the events that occur at the various 10.417 msec slots. In every slot, the 96Hz simulation tasks are executed. Every 6<sup>th</sup> slot, the 16Hz tasks are executed and every 96<sup>th</sup> slot the 1Hz tasks as well.

Each time the OBC writes at a particular VME memory address, the ScramNet reflective memory boards induce a hardware PCI interrupt on the RTS PC. This interrupt is transferred directly to EuroSim. EuroSim traverses through its schedule to execute the corresponding tasks, and writes the results to the PCI ScramNet board in the RTS. These results must be transferred to the OBC in time to guarantee real-time performance.

In the picture, the  $\Delta T$  indicate the margin for real-time execution, which limits the amount of allowable jitter over the end-to-end interface.

Several tests have been performed to verify and validate the RTTB:

- Open-loop tests, to verify correct transition from Matlab to EuroSim.
  - Observed relative deviations  $\pm 1E-6$  (insignificant).
- Real-time tests, to verify EuroSim slaving to OBSW at 96Hz closed-loop.
  - Established maximum jitter of  $\pm 0.20$  msec, well below acceptable threshold.
- Interface tests:
  - RTS – OBC interface test,
  - RTS – T3X interface test,
  - TM/TC PC – OBC interface test,
  - OBC - CMG interface test (performed by Deimos).
  - Solved various problems with drivers and protocols.
- Closed-loop tests (without T3X):
  - short duration acceptance test,
  - long duration acceptance test.
  - No major problems found.

The following picture shows a screenshot of the EuroSim test controller interface, showing a simulation run with the OBC as hardware-in-the-loop. As can be seen, the CPU load of the simulation is low enough to confide in the real-time performance of the RTS computer.

The message log panel shows a warning that the OBC is found out-of-sequence during initialisation, which is normal as the OBC continuously generates the 96Hz interrupts even when EuroSim has not been started. During initialisation both RTS and OBC sequence counters are synchronised.

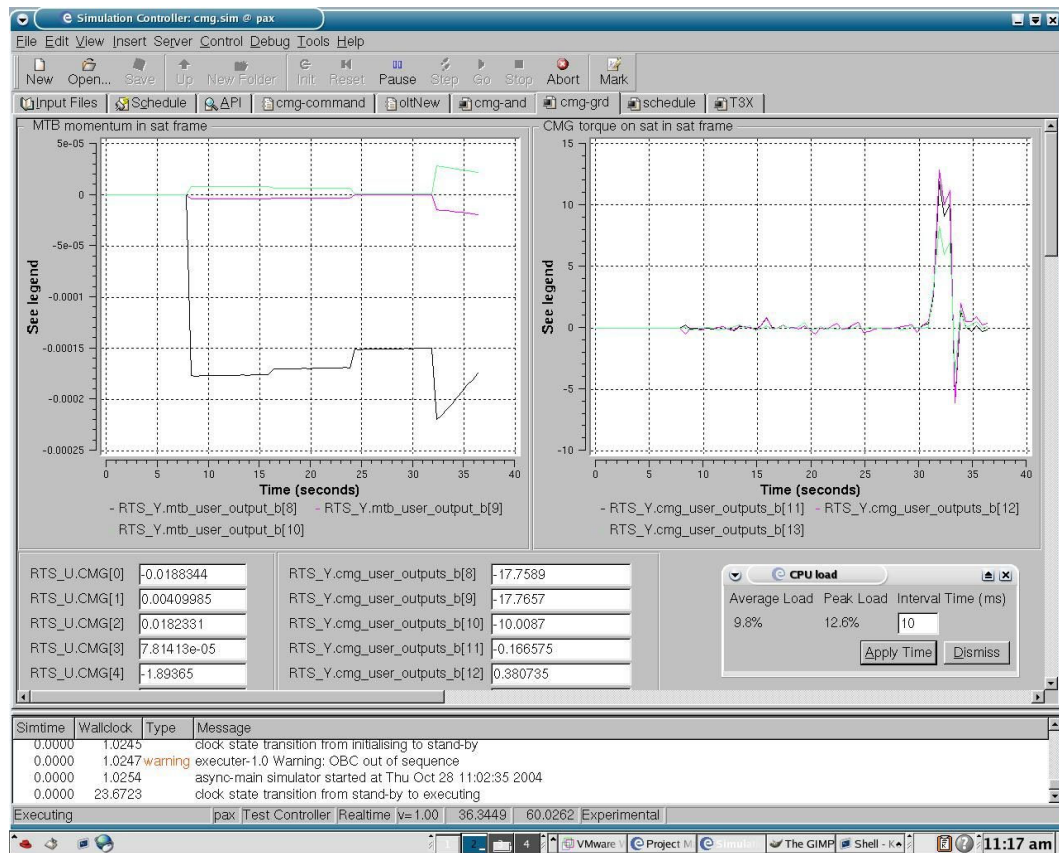


Fig. 10 RTTB example simulation run

For acceptance of the RTTB, two tests are defined with corresponding functional success criteria. It is found that the RTTB respects the success criteria of open-loop & closed-loop reference runs, in that no curve superposition was required and the resulting CMG determinant curve matched the one obtained from pure (Matlab) simulations.

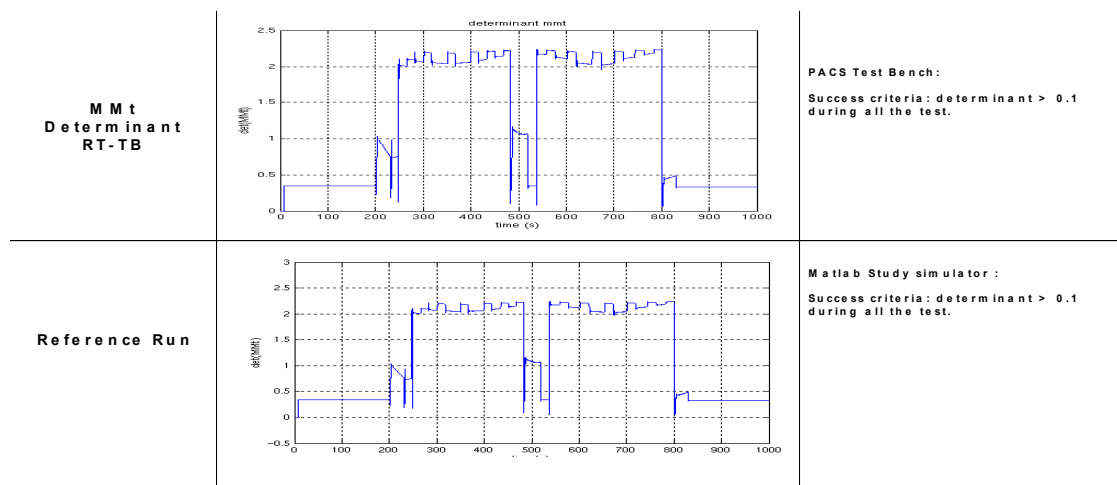


Fig. 11 RTTB comparison of results with and without hardware-in-the-loop



Some problems occurred during the development of the RTTB:

- simulation model transfer from Matlab to EuroSim via Mosaic is easy; however, multi-frequency blocks caused scheduling problems; the simple solution was to execute all blocks at 96Hz, except for the 8 Hz interface to rotation table;
- EuroSim had to be modified to cope with 96 Hz and intervals of non-integer milliseconds (10.417 msec);
- ScramNet driver had to be modified to transfer interrupts directly to EuroSim;
- an SGI IRIX 6.5 patch was necessary to fix an operating system bug with the thread priority switching mechanism;
- Tharsys ERC32 VME board needed modifications for the monitor software to handle EEPROM loading and for the serial port to run at 115 kbaud.

## 5 Conclusion

The development of the ACMS simulation, On-Board Software, and Real-Time Test Bench resulted in the successful achievement of a test bench, operating with hardware-in-the-loop at a closed-loop frequency of 96Hz. Other achievements:

- proven Matlab RTW autocoding for smooth development of OBASW and RTS;
- relatively easy development of a HIL RTTB using EuroSim;
- functional RTTB ready for testing CMG-in-the-loop on top of a 3-axes rotation table to account for gyroscopic effects due to the (satellite) motion;

The third and last phase of the PACS project began after delivery of the accepted RTTB to Alcatel Space premises in Cannes, France. In this phase, the test bench is used in a set-up with both a CMG breadboard in-the-loop and the OBSW running on the ERC32 in-the-loop to execute the following test:

TEST #5 "NOM: succession of 2 BRDF sequences"

- Sun pointing phases.
- Rallying phase with a S/C body rate greater than 3 deg/s.
- First BRDF sequence with 35 deg roll.
- Rallying phase.
- Second BRDF sequence with -20 deg roll.
- Rallying phase.
- Sun pointing phases.



## 6 Acknowledgements

NLR thankfully acknowledges the fruitful co-operating with Alcatel Space and Deimos for their co-operation and their contributions to this paper. Also the useful feedback from ESA for their suggestions in improving the PACS Real-Time Test Bench is much appreciated.

## 7 References

1. Ingen Schenau, H.A. van; Rijn, L.C.J. van; Spaa, J., *Test and Verification Equipment for the Attitude & Orbit Control System of the XMM satellite*, NLR-TP-98236, Athens, DASIA 1998.
2. Brouwer, M.P.A.M.; Casteleijn, A.A.; Ingen Schenau, H.A. van; Oving, B.A.; Timmermans, L.J.; Zwartbol, T., *Developments in Test and Verification Equipment for spacecraft*, Noordwijk, SESP 2000.
3. Timmermans, L.J.; Zwartbol, T.; Oving, B.A.; Casteleijn, A.A.; Brouwer, M.P.A.M., *From Simulations To Operations: Developments In Test And Verification Equipment For Spacecraft*, NLR-TP-2001-323, Noordwijk, DASIA 2001.
4. Lammen, W.F.; Nelisse, A.H.W.; Dam, A.A. ten, *Mosaic: Automated Model Transfer In Simulator Development*, Noordwijk, SESP 2002.
5. Eurosim, [www.eurosim.nl](http://www.eurosim.nl)