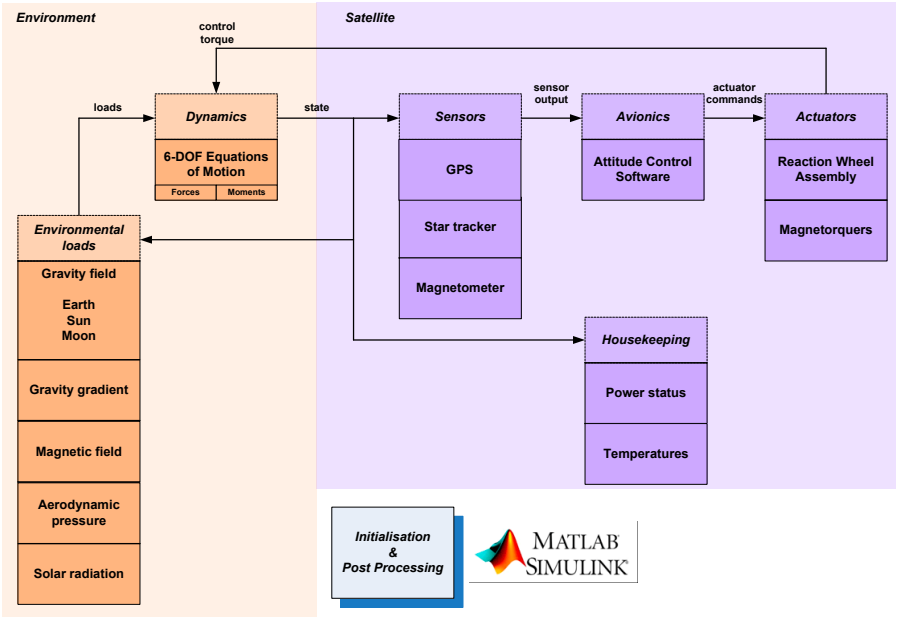
**Nationaal Lucht- en Ruimtevaartlaboratorium**
National Aerospace Laboratory NLR

## Executive summary

# Real-Time AOCS EGSE Using EuroSim and SMP2-Compliant Building Blocks




**Report no.**
NLR-TP-2012-394

**Author(s)**
B.A. Oving
A.J.P. van Kleef

**Report classification**
UNCLASSIFIED

**Date**
February 2013

**Knowledge area(s)**
Ruimtevaartinfrastructuur

**Descriptor(s)**
EGSE
AOCS
SMP2
on-board software
test bench

### Problem area

This paper focuses on lessons learnt about embedded simulator model development and conversions to organise a real-time architecture in the simulation environment EuroSim.

Attention is paid to obtain a conveniently arranged data dictionary in Eurosim for plotting and scripting, to support the development and testing of on-board software.

### Description of work

A spacecraft environment and a satellite model including sensors, on-board software for attitude control and actuators are developed in Embedded Matlab/Simulink.

These models are then converted into SMP2 standard models on unit level using Matlab's Real Time Workshop (RTW) module and the Model-Oriented Software Automatic Interface Converter (MOSAIC).

This report is based on a presentation held at the SESP 2012, ESTEC Noordwijk, September 26, 2012.

The SMP2 models are imported into EuroSim, where simulation data is graphically visualised by OpenSceneGraph and Satellite Tool Kit to support system engineering activities.

A hardware platform is set up to act as a closed-loop test bench to verify the on-board software running on the target platform.

**Results and conclusions**

The conversion steps, from model development in Matlab/Simulink into a real-time SMP2 model based simulation environment were successfully performed using RTW and MOSAIC.

The automated coding result of the SMP2 model for the attitude control software proves too difficult to modify for operational on-board software. Therefore, the software was rewritten in C/C++.

Asynchronous execution of the attitude control algorithms shows little deviation of the control algorithms running synchronously.

The process for developing attitude control software is able to provide good results with small effort.

**Nationaal Lucht- en Ruimtevaartlaboratorium**, National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 88 511 31 13, Fax +31 88 511 32 10, Web site: www.nlr.nl

**Nationaal Lucht- en Ruimtevaartlaboratorium**
National Aerospace Laboratory NLR

NLR-TP-2012-394

# Real-Time AOCS EGSE Using EuroSim and SMP2-Compliant Building Blocks

B.A. Oving and A.J.P. van Kleef

This report is based on a presentation held at the SESP 2012, ESTEC Noordwijk, September 26, 2012.



| | |
|---|---|
| Customer | National Aerospace Laboratory NLR |
| Contract number | ----- |
| Owner | National Aerospace Laboratory NLR |
| Division NLR | Aerospace Systems |
| Distribution | Unlimited |
| Classification of title | Unclassified |
| | February 2013 |

Approved by:

| Author | Reviewer | Managing department |
|---|---|---|
| B. A. Oving | A.J.P. van Kleef | P. Dieleman |
| Date: 26.02.2013 | Date: 26-02-2013 | Date: 26 - 2 - 13 |

## Summary

This paper describes a real-time attitude control simulator architecture in EuroSim using SMP2 based simulator models towards a future EGSE system for AOCS hardware-in-the-loop testing activities.

A spacecraft environment and a satellite model including sensors, on-board software for attitude control and actuators are developed in Embedded Matlab/Simulink. Consequently, these models are converted into SMP2 standard models on unit level using Matlab's Real Time Workshop (RTW) module and the Model-Oriented Software Automatic Interface Converter (MOSAIC). In EuroSim, simulation data is graphically visualised by controlling OpenSceneGraph and Satellite Tool Kit to support system engineering activities.

This paper focuses on lessons learnt about embedded simulator model development and conversions to organise a real-time architecture in EuroSim, with a conveniently arranged data dictionary for plotting and scripting, to support the development and testing of on-board software.

# Contents

## Abbreviations

| | |
|---|---|
| COTS | Commercial Off-The-Shelf |
| DOF | Degrees-Of-Freedom |
| EGSE | Electrical Ground Support Equipment |
| ENV | Environment |
| GPS | Global Positioning System |
| HIBRIS | Highly Integrated Broadband Imaging Spectrometer |
| LAN | Local Area Network |
| MCL | Magnetic Coils |
| MGM | Magnetometer |
| MOSAIC | Model-Oriented Software Automatic Interface Converter |
| OBS | On-board Software |
| POW | Power subsystem |
| RTW | Real-Time Workshop |
| RWS | Reaction Wheel System |
| SMP | Simulation Model Portability |
| STR | Star Tracker |
| TCS | Thermal Control Subsystem |

# 1 Introduction

Typically, a real-time simulator for EGSE activities contains various sub-models from different suppliers. The SMP2 standard proposed by ESA can be used to interchange models in various simulators, thus enabling efficient re-use. The integration of SMP2 models into the EuroSim simulation environment requires a structured process.

An example scenario targeting attitude control of a small satellite was used to evaluate the conversion process from Matlab simulations to real-time EGSE. First, a spacecraft environment and a satellite model (including on-board software for attitude control) were developed in Matlab/Simulink. Consequently, these models were converted to C using Matlab's Real Time Workshop (RTW) module. MOSAIC then converts the C code into SMP2 models, ready for integration into EuroSim.

In the next step, the attitude control functionality was rewritten from the embedded Matlab scripts into the C/C++- programming language as part of the operational on-board software development. In order to test/debug the attitude control, this software was written such that it can also run as an additional model inside EuroSim together with the original SMP2 models.

Finally, after successful validation steps, the on-board attitude control software is executed on the target on-board computer, where data is exchanged with the simulator via a local area network.

These above steps will be detailed further in this paper:

1. Spacecraft environment and satellite model development in Matlab/Simulink
2. Conversion of all Matlab models into SMP2 models and EuroSim files using MOSAIC
3. Conversion of the Matlab attitude control model into operational C/C++ on-board software
4. Real-time attitude control simulation in EuroSim including both the SMP2 attitude control model and the operational on-board software for validation
5. Separation of EuroSim simulator running on test bench computer and on-board software running on the target on-board computer

# 2 Simulator Architecture

As part of NLR's research programme, a small satellite (20 kg) was designed targeting an Earth observation mission [1] using a hyperspectral imager, based on the Highly Integrated Broadband Imaging Spectrometer (HIBRIS) from cosine. In this mission scenario, the satellite requires precise-pointing to nadir from a polar orbit at 500 km altitude.

After release from the launcher, the satellite must be de-tumbled first with course nadir-pointing using magnetometers and coils. Thereafter, the satellite enters precise-pointing mode using a GPS, star tracker and a reaction wheel assembly.

A simulation of this satellite was developed in Matlab/Simulink (R2010b 64-bit version) with models for the attitude control equipment and with a model for attitude control.

Figure 1 shows the top-level simulator architecture. The architecture contains an environment model (ENV) for the six degrees-of-freedom (6-DOF) dynamics, disturbances, and frame transformations. Models for dynamics, disturbance and frame transformations are based on GGNCSIM library models of Dutch Space (DS). These models are described in [2] and are based on Matlab/Simulink S-Functions.

The following disturbance forces are taken into account in the environment model:

- Gravity gradient (torque)
- Magnetic (torque)
- Solar radiation pressure (torque and force for 6 sides of a cubed sized satellite)
- Aerodynamic drag (torque and force for 6 sides of a cubed sized satellite)
- Third body accelerations for orbit by Sun and Moon
- Earth's gravity effects for orbit (user choice for central, zonal harmonics or GRIM5C)

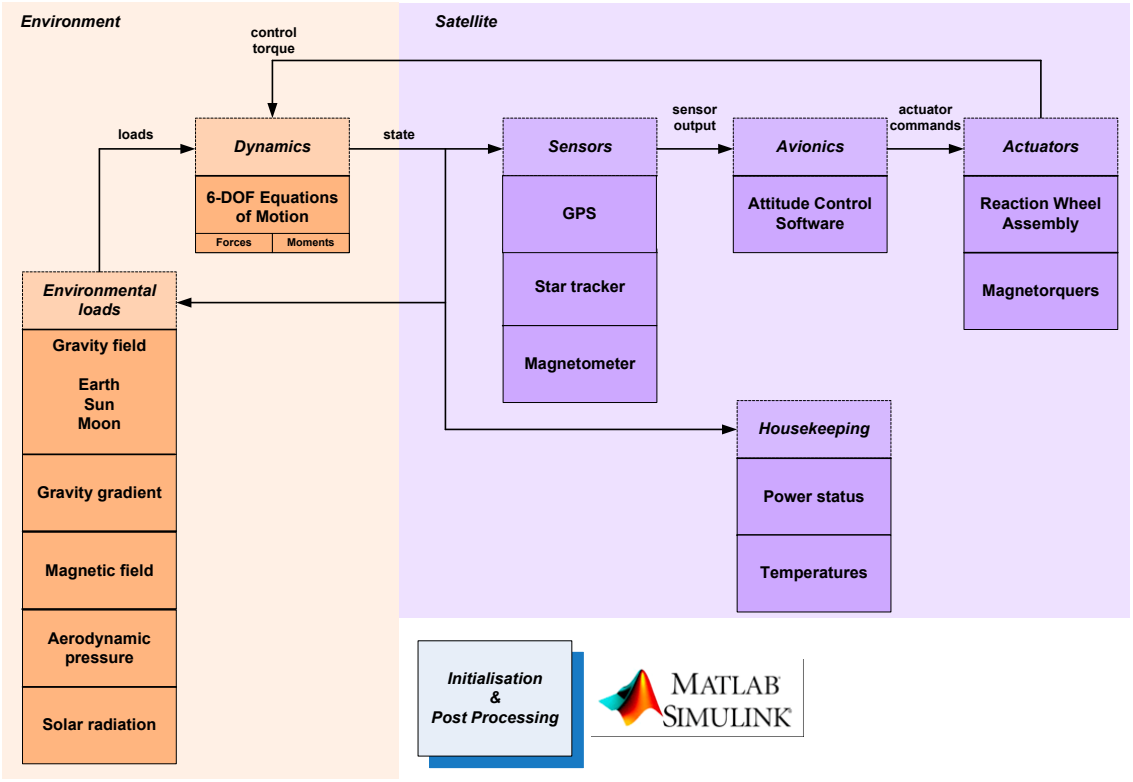



*Figure 1 Simulator architecture in Matlab/Simulink*

The satellite model consists of sensors, actuators and on-board software for attitude control:
Sensors:

- Global Positioning System (GPS) receiver
- Magnetometer (MGM)
- Star Tracker (STR) including noise and blinding functionality

Actuators:

- Magnetic Coils (MCL) about three body axis
- Reaction Wheel System (RWS) about three body axis including friction and stiction

Attitude control software for three attitude control modes:

- De-tumbling (MGM and MCL)
- Intermediate nadir pointing (GPS, MGM, and MCL)
- Precise nadir pointing (GPS, STR, and RWS)

Besides attitude control, power generation and consumption as well as temperature estimation of various satellite nodes are part of the satellite models. The models are a mix of Matlab S-Functions (environment) and Embedded Matlab code (satellite).

# 3 Simulator SMP2 Model Generation

The conversion from simulator models developed in Matlab/Simulink into SMP2 standard models in EuroSim required two functional steps.

The first step is using RTW ("Build Subsystem…" option) in Matlab/Simulink for code generation of the 9 sub-models: ENV, GPS, MGM, STR, OBS, MCL, RWS, POW and TCS. The MOSAIC manual [13] was used to set the correct RTW settings in the Simulink file. These sub-models are modified in Matlab/Simulink to support a clearer model Input/ Output (I/O) parameter structure in the Data Dictionary of EuroSim. Input and output of sub-models are established by Matlab/Simulink signals and parameters listed in the Embedded Matlab code function arguments.

The second step is to convert these sub-models into the SMP2-compliant models and EuroSim project files. MOSAIC (Model-Oriented Software Automatic Interface Converter, version 9.01) automates model transfer between Commercial Off-The-Shelf (COTS) tools and simulation standards. The intended use of MOSAIC is for developers of systems who wish to benefit from the potential modelling capabilities of MATLAB/Simulink/Stateflow, EcosimPro, and 20-sim, the potential of the simulation shells of EuroSim/SIMSAT/Basiles, and/or the potential of ESA's SMP2 standard.

In EuroSim, the model inputs and outputs were interconnected using EuroSim's Model Description and Parameter Exchange Editors to demonstrate the integration of multiple SMP2 models originating from different suppliers.

# 4 On-Board Software Development

The operational on-board software is targeted for a low cost on-board computer, suitable for the reference mission, which features a powerful Cortex-A9 processor at only 1 W power consumption. This computer board has a form factor that can be applied in CubeSat, although not PC104. The Cortex-A9 SoC allows failure mitigation techniques such as Triple Modular Redundancy (TMR) for dealing with soft errors (bit flips) to be implemented in software, instead of in hardware.

A failure-tolerant software application framework was developed in C under Linux, which applies various mitigation techniques:

- Keep-alive mechanism to monitor other processes
- Checksum mechanism to check own code integrity
- Redundancy in data (triple data with voting and scrubbing)
- Redundancy in processes (triple processes with voting)

Figure 2 presents an overview of this framework, where processes are launched that are responsible for the actual processing of the functional software, such as the attitude control software. Inter-process communication is done via shared memory (shm). All data is protected by triple modular redundancy, whenever possible. This applies to internal variables and to external variables in shared memory.

More than one instance of a process can be launched by forking, thus implementing redundancy in processes. The results of these processes are combined (via voting) by another process.

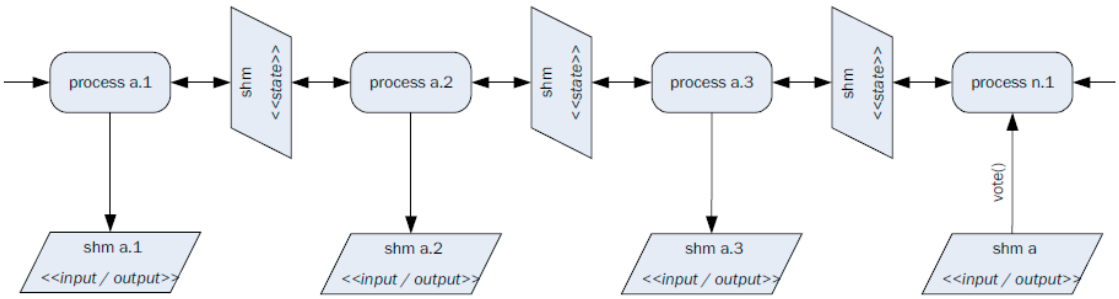



*Figure 2 Failure-tolerant application framework*

The processes implement a keep-alive mechanism that uses a state flag, which is set by the one process to be unset by the other. Periodically, it is checked if the state indeed has changed. The

same mechanism is used to control the state of execution of processes and their children. In this way, synchronisation between the processes (specifically, the children) can be enforced.

Furthermore, each process periodically checks the integrity of its code segment. If the check fails, the keep-alive flag is not updated: the process is killed and a new process is started.
The attitude control software as developed in Embedded Matlab functions was initially converted by RTW as an SMP2-compliant model for use in EuroSim.

However, the automated code generation lead to poor readability and (more important) adaptability of the resulting software, which proved too difficult to integrate into the software application framework. Therefore, the Embedded Matlab functions were rewritten in C++ to enable easy integration into the operational on-board software framework.

The SMP2 model for attitude control was then used for validation of the resulting operational software: both attitude control models (SMP2 and operational) were executed simultaneously in EuroSim to validate the conversion from Matlab to C++.

As expected, after some debugging, the results were identical. This shows that this process for developing attitude control software is indeed viable and is able to provide good results with small effort. In fact, most of the effort was to develop the complete Matlab simulator. The transition to EuroSim was a matter of a few hours, whereas the conversion to C++ took a few man weeks.

# 5 On-Board Software and Test Bench

The next step was to integrate the attitude control software into the on-board software framework and execute it on the target on-board computer, while the satellite and the dynamics are simulated on a separate test bench computer (Fig. 3.). A LAN interface between the platforms was used for data transfer.

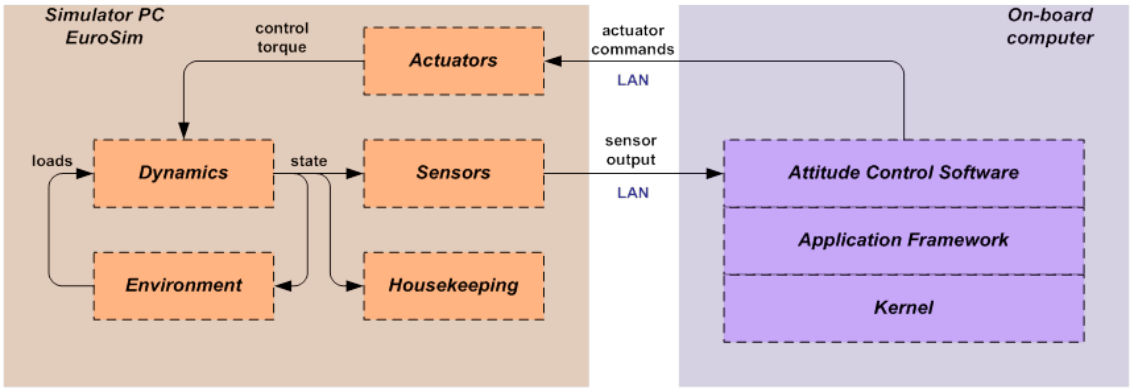



*Figure 3 Test bench setup*

Before, the complete simulation was executed at 2 Hz, which is the frequency of the attitude control algorithm. Now that the attitude control is executed on a different platform, the simulator executes the models at 20 Hz and the data exchange between the platform runs at 4 Hz.

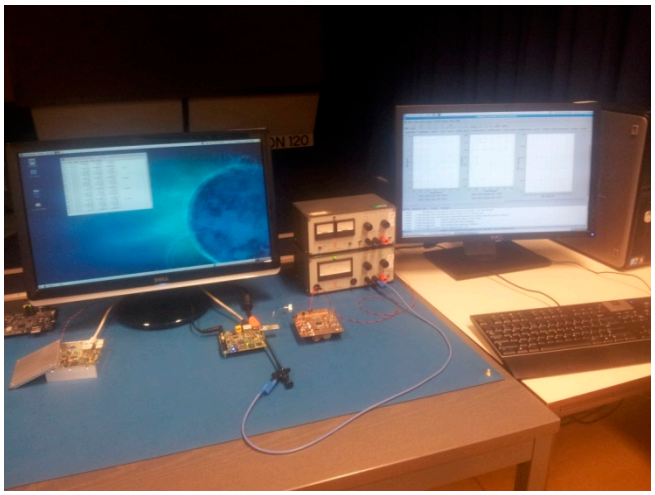


*Figure 4 Test bench with target computer on the left and simulator on the right*

# 6 Conclusions

The conversion steps, from model development in Matlab/Simulink into a real-time SMP2 model based simulation environment in EuroSim, were successfully performed using RTW and MOSAIC.

The automated coding result of the SMP2 model for the attitude control software proves too difficult to modify for operational on-board software. Therefore, the software was rewritten in C/C++.

The SMP2 model is still useful for debugging and validation of the operational software by embedding both models in the same EuroSim simulation environment (both running in parallel). A hardware platform has been set up to act as a closed-loop test bench to verify the on-board software running on the target platform. Asynchronous execution of the attitude control algorithms shows little deviation of the control algorithms running synchronously.

The followed process for developing attitude control software is able to provide good results with small effort.

## References

[1] B.A. Oving, A.J.P. van Kleef, *Design of a Small Satellite for a HyperSpectral Instrument*, paper presented at the 4S symposium 2012, NLR-TP-2012-292

[2] M.H.M. Ellenbroek, *Generic GNC Simulator Environment - Detailed Design of the GGNCS*, GGNCS-DS-SP-003_DDD, 2007

[3] W.F. Lammen, *Automated model transfer from MATLAB R2010b/Simulink, EcosimPro and 20-sim to ESA's Simulation Model Portability SMP2 standard and the real-time simulation engine EuroSim - MOSAIC 9: User Manual*, NLR-CR-2011-475