Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 11245–11250

The Design and Implementation of the UPMSAT-2 Attitude Control System [⋆]

Juan Zamorano, [*] **Jorge Garrido,** [*] **Javier Cubas** [**]
Alejandro Alonso, [*] **Juan A. de la Puente** [*]

[*] *Information Processing and Telecommunications Center (IPT)*
[**] *Instituto Universitario de Microgravedad "Ignacio da Riva" (IDR)*
Universidad Politécnica de Madrid (UPM), Spain

Abstract: The paper describes in detail the design and implementation of the attitude control system of a small satellite, focusing on the hardware and software aspects. A new control algorithm has been used in order to deal with particular requirements of the system, which are related to its communication, power generation, and thermal control needs. The system is implemented on an embedded on-board computer, and the software has been developed with a model-driven approach using state-of-the-art software tools. Simulation tools have been used for software testing and validation in an integrated validation facility.

Keywords: Embedded computer control systems and applications; Logical design, physical design, and implementation of embedded computer systems; Micro-nano-aerospace vehicles/satellites

1. INTRODUCTION

Attitude control is a key element of all kinds of spacecraft. Keeping an appropriate orientation with respect to the Earth or other celestial bodies is a basic requirement for a correct antenna orientation, thermal control, and power generation, among others (see e.g. Fortescue et al., 2011).

In this paper we describe the overall design and implementation of the attitude control system (ACS) of UPMSat2, a micro-satellite under development at IDR-UPM with the collaboration of the the STRAST group for the on-board and ground segment software.

The satellite has an envelope of 0.50 × 0.50 × 0.60 m, with an approximate mass of 50 kg (figure 1). The lateral faces, and part of the top face, are covered with solar panels. A dipole antenna for radio communication with the ground segment is located on the top face. The satellite will be set on a noon/midnight sun-synchronous orbit with an altitude of 700 km and a period of 98 min. The eclipse time is 36 min.

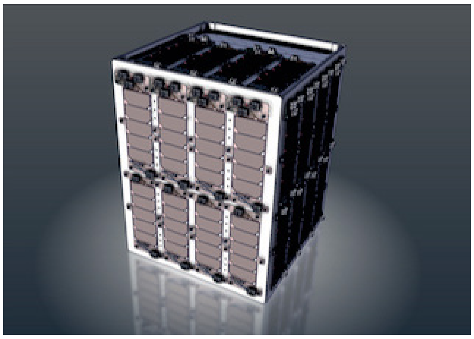

Fig. 1. General view of the satellite platform

The UPMSat-2 payload consists of a set of experiments aimed at demonstrating some new technologies for spacial use. There are no image acquisition cameras or any other subsystems that may require an accurate control of the attitude. However, visibility of the radio antenna from the ground stations must be ensured at all times. A simple way of achieving this purpose is to keep the antenna normal to the orbital plane. Other important requirements are to balance the amount of solar radiation received by the side panels, and to keep temperature distribution within the satellite as uniform as possible. A slow rotation around the tangential direction of the orbit is a simple, effective way of achieving these goals.

Previous articles on the UPMSat-2 software focused on the use of model-driven engineering methods (de la Puente et al., 2014b,a), real-time analysis (Garrido et al., 2012, 2013; Zamorano and Garrido, 2015), and alternative partitioned architectures (Salazar et al., 2014; Alonso et al., 2015). The attitude control algorithm used in UPMSat-2 was introduced by Cubas et al. (2015). This paper concentrates on the design of the attitude control system form and its implementation on the UPMSat2 on-board computer, including hardware and software aspects and system validation. Section 2 presents the overall approach to attitude control and the control algorithms used on the satellite. Section 3 describes the implementation of the attitude control system and the software solutions that have been put in place for the problems encountered. The approach used for validating the system is described in section 4. Finally, section 5 contains some conclusions and hints for future work.

[⋆] This work has been partially funded by the Spanish National R&D&I plan (projects M2C2, TIN2014-56158-C4-3-P).

10.1016/j.ifacol.2017.08.1607

2. ATTITUDE CONTROL

2.1 Basic concepts

The attitude of a satellite can be expressed in terms of the orientation of the satellite with respect to an Earth-based reference frame. The following reference frames are commonly used (figure 2):

- Body reference frame (X, Y, Z), with its origin in the satellite centre of mass. The axes are oriented according to the principal inertial axes, with the Z axis in the direction of the minimum moment of inertia.
- Orbital reference frame (X_0, Y_0, Z_0), with its origin in the centre of the Earth. The X_0 axis pointing toward the ascending node of the orbit, the Z_0 axis perpendicular to the orbit toward the positive direction of the rotation axis of the satellite in its orbit, and the Y_0 axis completing the right-hand Cartesian frame.

  It is worth noting that on some orbits, in particular for sun-synchronous orbits with a low precession rate, this reference frame can be considered inertial for short periods of time.

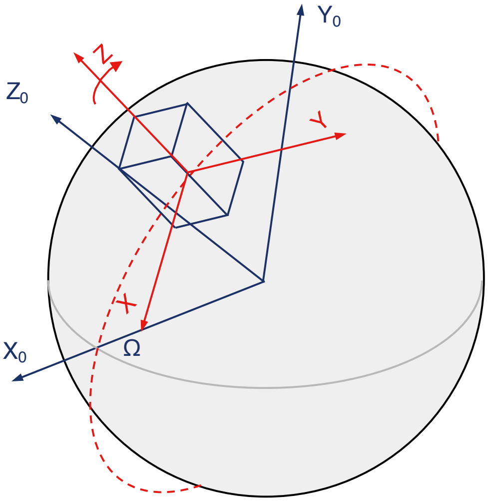

Fig. 2. Reference frames.

The aim of the attitude control system is to generate a control torque such that the control objectives are met. Generally speaking, this implies establishing some relationship between the body frame and the orbital frame, which can be characterized in terms of a transformation matrix.

The dynamics of the satellite can be derived from Euler's equation (de Ruiter et al., 2012):

$$\mathbf{I}\dot{\omega} + \omega \times \mathbf{I}\omega = \mathbf{T} \tag{1}$$

where ω is the angular velocity vector with respect to the orbital frame, $\mathbf{I}$ is the inertia matrix, and $\mathbf{T}$ is the vector torques actuating on the satellite body, which can be decomposed as

$$\mathbf{T} = \mathbf{T}_c + \mathbf{T}_d \tag{2}$$

where $\mathbf{T}_c$ is a control torque, generated by the control actuators, and $\mathbf{T}_d$ is the disturbance torque generated by the environment.

There are different kinds of sensors that can be used for attitude determination, including sun sensors, horizon sensors, star trackers, and magnetorquers. Common attitude actuators include thrusters, reaction wheels, and magnetic torquers, or magnetorquers for short. Depending on the kind of sensors and actuators chosen for a particular system, a variety of control strategies for attitude control can be used in a space mission (Sidi, 1997). Figure 3 shows a scheme of an attitude control system using magnetic sensors and actuators, which is commonly known as magnetic attitude control.

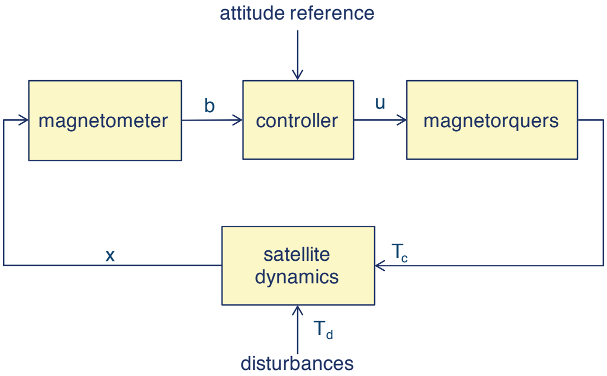

Fig. 3. Magnetic attitude control system.

2.2 Magnetic attitude control

Magnetic attitude control, based on the interaction of the Earth's magnetic field with the spacecraft, is a simple, reliable, and cost-effective approach often used in small satellites on low Earth orbits (LEO). Magnetometers and magnetorquers are commonly used as sensors and actuators for magnetic control. Magnetometers are magnetic sensors that provide a measurement of the strength and direction of the magnetic field, i.e. the magnetic field vector, at a given point. It should be noted that attitude determination from magnetometer readings requires using a mathematical model of the Earth's magnetic field. However, some control methods have been devised that use only magnetic field measurements and yet provide some degree of attitude angle control without entailing any such models (Sidi, 1997).

A magnetorquer is a magnetic coil which produces a magnetic moment that interacts with the Earth's field, thus enabling the attitude of the satellite to be changed. Three orthogonal magnetorquers are commonly used to produce a magnetic moment vector in an arbitrary direction. The resulting control torque can be expressed as

$$\mathbf{T}_c = \mathbf{m} \times \mathbf{b} \tag{3}$$

where $\mathbf{m}$ is the magnetic moment vector of the magnetorquers and $\mathbf{b}$ is the Earth's magnetic field vector.

The above equation shows that the actuating torque is always perpendicular to the Earth's magnetic field, which means that only partial attitude control is possible. However, if the direction of the magnetic field changes along the orbit, as is the case for sun-synchronous orbits, it is possible (at least on average over the orbital period) to apply actuating torques in any direction (Cubas et al., 2015).

A variety of control methods, both linear and nonlinear, can be used for magnetic attitude control (Silani and Lovera, 2005). A simple, robust control law is the so-called b-dot control law, which can be formulated as:

$$\mathbf{m} = -k\dot{\mathbf{b}} \tag{4}$$

This control law has been shown to converge to a state with a low angular momentum, making it suitable for the initial detumbling phase (Lovera, 2015). However, the final attitude and angular velocity cannot be set to a desired value, and therefore b-dot control cannot be used for steady state attitude control. A number of modifications have been proposed in order to overcome this limitation (Cubas, 2015). However, most of them require additional hardware or complex analytic models of the Earth's magnetic field, thus departing from the simplicity of pure magnetic control.

2.3 The UPMSat2 attitude control algorithm

In order to improve the performance of magnetic control without incurring the problems of current b-dot methods, a new control law has been developed by Cubas et al. (2015) to be used in the UPMSat-2. The control law is

$$\mathbf{m} = -k(\dot{\mathbf{b}} + \boldsymbol{\omega}_d \times \mathbf{b}) \tag{5}$$

where $\boldsymbol{\omega}_d$ is the desired angular velocity vector. This control law has been shown to be stable and robust. Moreover, after initial detumbling the angular velocity converges to $\omega = \omega_d$, and the spin rotation vector $\boldsymbol{\omega}_d$ tends to align with the direction of the $\dot{\mathbf{b}} \times \mathbf{b}$ vector, which for sun-synchronous orbits, with a high inclination, stays close to the Z_0 axis of the orbital reference frame. This is coherent with the UPMSat-2 attitude control requirements, where $\boldsymbol{\omega}_d$ is set to $[0\ 0\ \omega_r]^\mathsf{T}$ in the body frame, thus making the desired spin rotation axis parallel to the body Z axis.

Three consecutive stages can be identified: a detumbling stage, in which the X and Y components of angular velocity tend to zero and the Z component tends to its desired value, while the angle between the Z and the Z_0 axes is not controlled; a second stage in which the satellite rotation axis gets close to be aligned with the Z_0 axis; and a final stage in which the nominal conditions (ω_d and Z axis alignment) are maintained by the control system. Simulation results show good expected results for this control law in all these stages.

It should be noted that this control law does not require the attitude of the satellite to be computed on board, as it uses only magnetic field measurements. The attitude angles can be computed on ground using these measurements, together with solar cell data, which are periodically downloaded in telemetry messages.

3. THE UPMSAT-2 ATTITUDE CONTROL SYSTEM

3.1 Control system architecture

The attitude control method described in section 2 is implemented with a set of components, which are organised as shown in figure 3 above.

Magnetometers. A redundant set of two identical flux-gate magnetometers is used as the main sensor for control. A third magnetometer, from a different manufacturer, is available as an additional source of data. Each magnetometer yields three analog voltage signals, one for each component of the local magnetic field vector in the body reference frame. The signals are acquired by the on-board computer system by means of a 12-bit analog-digital converter (ADC). Conversion to engineering units (nT) and signal filtering is carried out by software.

Magnetorquers. Three single-axis magnetorquers generate the three components of the actuating magnetic moment vector on the body frame axes. The magnetorquer coils are commanded by means of a software-controlled pulse width modulation (PWM) signal. Since the control field may have to act on either positive or negative direction, an H-bridge commanded by two digital outputs is used to generate the actuating signal for each magnetorquer.

On-board Computer. The control algorithm, as well as the signal conditioning and other operations required for attitude control, are executed on the UPMSat-2 on-board computer (OBC). The OBC is based on a LEON3 (Gaisler, 2012) single core processor running at 20 MHz with 4 MB SRAM memory and 2 MB EEPROM memory. The OBC external interfaces include 64 analog input channels and 112 digital input/output signals, as well as an RS422 serial interface for connecting the radio equipment.

3.2 Control cycle

Magnetometers cannot be read while magnetorquers are being actuated upon, because the readings would be altered by the magnetorquers field. Therefore, each control cycle, with a duration of 2 s, is divided into two intervals. During the first half of the control cycle (1 s), the magnetometers are sampled with a period of 200 ms, and the sequence of average readings of the magntormeters is fed to the control algorithm, which takes a negligible time to compute the control action. The control action is output to the magnetorquers during the second half of the control cycle, which may take between 200 and 500 ms (figure 4). The final idle interval, with a minimum duration of 500 ms, prevents the magnetometers from undergoing magnetic interference from any residual magnetic moment in the magnetorquers.

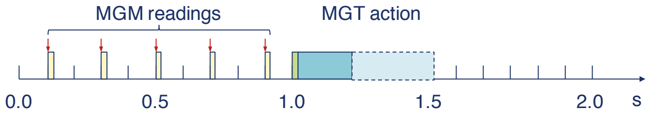

Fig. 4. Control cycle timing.

3.3 Software architecture

The on-board software system is composed of several subsystems, each providing a well-defined functionality to the operation of the spacecraft (figure 5). One of them is the ACS subsystem, which implements the attitude control functions.

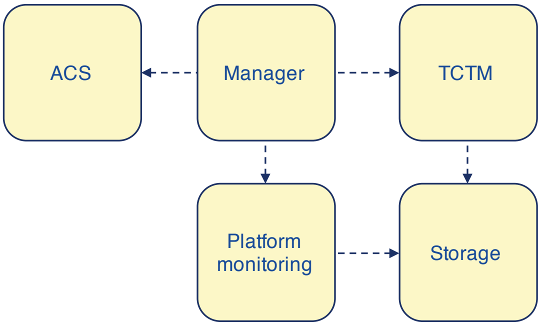

Fig. 5. UPMSat-2 software architecture.

The ACS high-level design consists of three concurrent tasks using two shared data objects for communication (figure 6). The task structure reflects that of the control system, dealing with sensor data acquisition, control algorithm calculation, and actuator command, respectively.

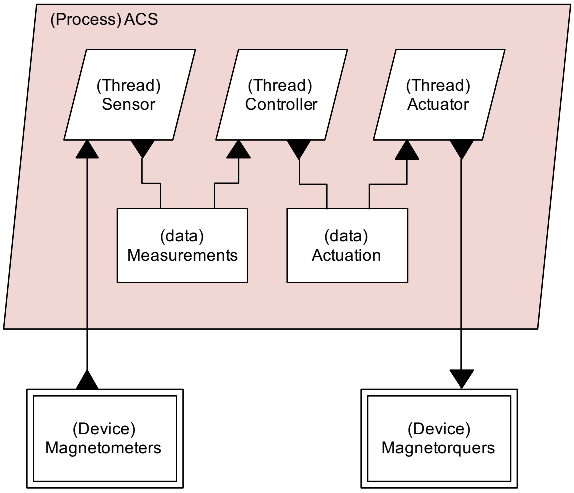

Fig. 6. ACS high-level design.

The *Sensor* task samples the analog inputs of the magnetometers according to the control cycle specification. The inputs are converted to engineering units using calibration data which are generated off-line for each magnetometer .

The *Controller* task implements the attitude control law as defined in equation 5, computing the control action to be output to the magnetorquers.

The *Actuator* task activates the PWM bridge for each magnetorquer according to the computed control action.

3.4 Software models

The UPMSat-2 software has been developed using a Model-Driven Engineering (MDE) approach (de la Puente et al., 2014a). Simulink models have been built for sensor signal conditioning, control algorithm, and other related operations.

Figure 7 shows the Simulink model of the controller. The main input (B_b_T) is the set of magnetometer readings from the *Sensor* task. The other inputs are parameters that can be changed by telecommands. The output of the model is the control signal for the magnetorquers, MT.

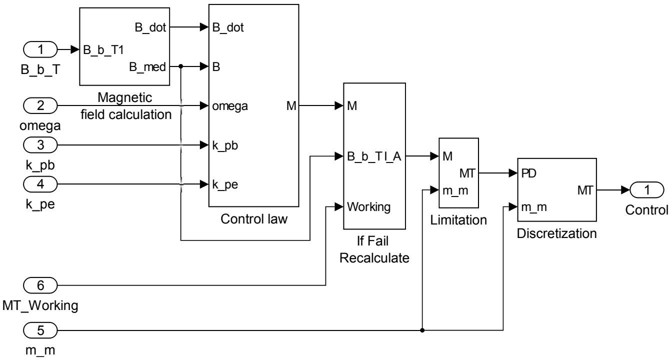

Fig. 7. Simulink model of the controller.

The first block, *magnetic field calculation*, provides an estimation of the magnetic field vector **b** and its derivative from the magnetic field measurements. It should be noted that the accuracy of the derivative depends on the precision in the duration of the sampling interval, which is related to the real-time behaviour of the sensor task (see section 3.6).

The next block, *control law*, implements equation 5. The block titled *if fail recalculate* outputs a degraded control action when one or two magnetorquers fail. Finally, the blocks titled *limitation* and *discretization* limit the actuation on the magnetorquers to its saturation value and produce the PWM duty cycle for the magnetorquer coils.

Data acquisition from the magnetometers in the *Sensor* task has also been modelled in Simulink. Its main functionality is to convert the readings of the three magnetometers to engineering units using calibration data, and store them into the Measurements data object.

3.5 Code implementation

C code for the controller and sensor algorithms has been automatically generated from their respective models using the Simulink code generator. The rest of the on-board software system has been manually written in Ada, or generated from other model-oriented languages. The Ada facilities for interfacing with C (ARM05) have been used to build a mixed-code system.

3.6 Real-time behaviour

All the UPMSat-2 on-board software subsystems have hard real-time requirements. In particular, the ACS activities must be synchronized in order to ensure that all the steps in the control cycle are executed in proper sequence and do not overlap with one another. For example, magnetometer measurements must not start before the specified delay following the end of magnetorquer actuation, in order not to be corrupted by residual magnetic fields. At a lower granularity level, the separation between consecutive readings of the magnetometers has to be implemented with a high precision, so that the derivative of the magnetic field can be accurately computed.

The on board software has been written using the restricted Ada Ravenscar tasking profile, in order to ensure that the real-time behaviour is predictable and analysable (ISO/IEC). The software runs on top the

ORK+ real-time kernel (de la Puente et al., 2000), which fully supports the Ravenscar profile. An extended form of schedulability analysis (Zamorano and Garrido, 2015) has been used to verify that the real-time behaviour of the system complies with the requirements.

4. ACS VALIDATION

4.1 Incremental validation approach

Software validation usually includes testing the system under real operating conditions. However, for obvious reasons, on-board space software cannot be tested in this way. In order to achieve a testing environment as close as possible to the space operating conditions, simulation models are commonly used. This is the approach that has been adopted in the UPMSat-2 project, where the variations of the Earth's magnetic field, the spacecraft dynamics and a number of disturbances have been incorporated in a comprehensive environment model. An incremental validation process has been defined, in which the ACS is first simulated together with environment model, and then the software is run on the OBC hardware interfacing with the simulation model. In the final validation stage the full satellite hardware is tested against the environment model.

4.2 Model-in-the-loop validation

The first validation phase uses a model of the ACS, together with models of the space environment and the spacecraft dynamics, to assess the validity of the control law and the design parameters (figure 8). Based on the model, a nominal sampling period of 2 s and a gain of $2 \cdot 10^8$ were found to be appropriate for stabilizing the attitude with a reference angular velocity of 0.1 rad/s on the Z axis.

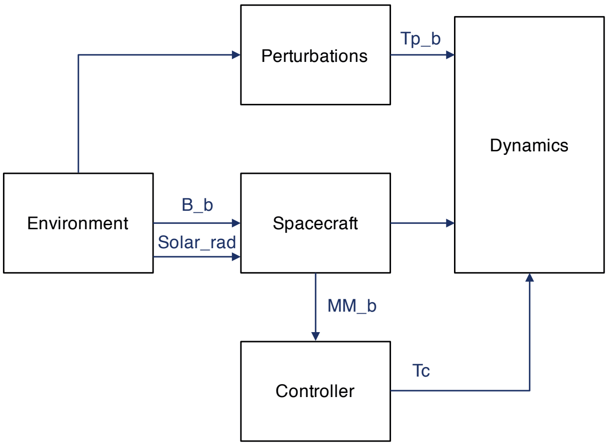

Fig. 8. UPMSat-2 ACS Simulink high level model view.

This kind of preliminary validation, usually called "model-in-the-loop" (MIL), can be used to validate the functional requirements of the system. However, in order to achieve full system validation all the elements of the space mission must be tested in an integrated way. The ultimate purpose of system validation process is to achieve a Technology Readiness Level (TRL) of "flight qualified".[1]

4.3 Processor-in-the-loop validation

A further step to full validation is running the control software on actual hardware, while still using simulation models of the sensors, actuators, and spacecraft dynamics. In order to do this, a software validation facility (SVF) has been deployed, which uses an auxiliary computer, linked to the OBC by a serial line, to run a simulation model of the Earth's magnetic field and the satellite dynamics. This model is basically the same as shown in figure 8, with the control law component of the *control* box replaced by the actual software running on the OBC. The values of the control variables are directly exchanged between the model and the computer in engineering units, thus not including a model of the sensors and actuators, which is deferred to the next step.

This setup is called "processor-in-the-loop" (PIL), and can be used to test the functional behaviour of the software implementation, as well as to analyse its temporal behaviour (Garrido et al., 2012). However, it does not allow the TRL to go beyond level 4.

4.4 Hardware-in-the-loop validation

The next step is to include a model of the actual magnetometers and magnetorquers in the simulation model, and connect their inputs and outputs to the OBC hardware by means of its analog and digital input and output lines. The refined model also includes the calibration data for the magnetometers and the three-step control cycle described in section 3.2.

This requires enhancing the SVF by adding analog and digital interfaces to the simulation computer. The *magnetic field* component of the model is modified so that it simulates the magnetometer output voltages, which are fed to the computer via ADC lines. Similarly, the actuator model takes as inputs the digital signals (PWM) output by the OBC, which drive the H-bridges of the simulated magnetorquers.

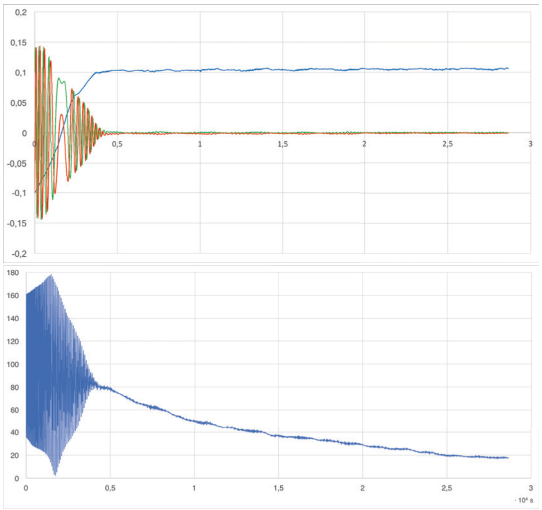

Fig. 9. Angular velocities and Z-Z_0 axis angle.

Figure 9 shows the results of a sample simulation run lasting almost 30000 s, equivalent to five orbits. The evolution of angular velocities and the angle of the Z axis

---

[1] Currently equivalent to TRL level 8 for ESA. `http://sci.esa.int/sci-ft/50124-technology-readiness-level`

in body reference axes for an angular velocity reference of $[0\ 0\ 0.1]^\mathsf{T}$ rad/s are displayed in the figure. The three consecutive stages defined in section 2.3 can be clearly identified. The detumbling stage ends at about $t = 4000$s. The alignment stage lasts until $t = 25000$s, after which time the Z axis is stabilised. See Cubas, Farrahi, and Pindado (2015) for more details on the performance of the control law and the tuning of the control parameters.

This simple kind of HIL setup allows the operation of input/output devices, as well as signal conditioning hardware and software, to be validated. For example, noise in the UPMSat-2 ADC channels could be characterized using this approach, which enabled the engineers to improve the design of this crucial part of the OBC interface hardware.

Full HIL validation, though, requires using the real magnetometers and magnetorquers hardware and, ultimately, the whole satellite structure, on a setup which enables the attitude of the spacecraft to evolve in a free fashion. Air-bearing testbeds have been used to this purpose (see e.g. Schwartz et al., 2003) and will be the subject of further assessment by the UPMSat-2 team.

5. CONCLUSIONS

The attitude control system of the UPMSat-2 microsatellite has been described. The spacecraft attitude is controlled using only magnetic sensors and actuators. The system features a new magnetic attitude control method which has been validated through extensive simulation, with results showing good control performance with low energy consumption, as required from this kind of satellites.

The control software has been developed using MDE methods, which has proven to be very useful for achieving a high modularity and an efficient software implementation. The software has been validated using an incremental model-based approach, which has been found to be both efficient and appropriate for this kind of systems.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the help and support of all the members of the IDR and STRAST teams.

REFERENCES

Alonso, A., de la Puente, J.A., Zamorano, J., de Miguel, M.A., Salazar, E., and Garrido, J. (2015). Safety concept for a mixed criticality on-board software system. *IFAC-PapersOnLine*, 48(10), 40 – 245.

ARM05 (2007). *ISO/IEC 8652:1995(E)/ TC1(2000)/ AMD1(2007): Information Technology — Programming Languages — Ada.*

Cubas, J. (2015). *Analytical models for the power subsystem and the attitude control subsystem of a microsatellite.* Ph.D. thesis, Universidad Politécnica de Madrid.

Cubas, J., Farrahi, A., and Pindado, S. (2015). Magnetic attitude control for satellites in polar or sun-synchronous orbits. *Journal of Guidance, Control, and Dynamics*, 38(10), 1947–1958.

de la Puente, J.A., Garrido, J., Zamorano, J., and Alonso, A. (2014a). Model-driven design of real-time software for an experimental satellite. *IFAC Proceedings Volumes*, 47(3), 1592 – 1598. 19th IFAC World Congress.

de la Puente, J.A., Ruiz, J.F., and Zamorano, J. (2000). An open Ravenscar real-time kernel for GNAT. In H.B. Keller and E. Plödereder (eds.), *Reliable Software Technologies — Ada-Europe 2000*, number 1845 in LNCS, 5–15. Springer-Verlag.

de la Puente, J.A., Zamorano, J., Alonso, A., Garrido, J., Salazar, E., and de Miguel, M.A. (2014b). Experience in spacecraft on-board software development. *Ada User Journal*, 35(1), 55–60.

de Ruiter, A., Damaren, C., and Forbes, J. (2012). *Spacecraft Dynamics and Control: An Introduction.* Wiley.

Fortescue, P., Swinerd, G., and Stark, J. (2011). *Spacecraft Systems Engineering.* Wiley, 4 edition.

Gaisler (2012). *LEON3 - High-performance SPARC V8 32-bit Processor. GRLIB IP Core User's Manual.* Gaisler Research.

Garrido, J., Brosnan, D., de la Puente, J.A., Alonso, A., and Zamorano, J. (2012). Analysis of WCET in an experimental satellite software development. In T. Vardanega (ed.), *12th International Workshop on Worst-Case Execution Time Analysis*, volume 23 of *OpenAccess Series in Informatics (OASIcs)*, 81–90. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Garrido, J., Zamorano, J., and de la Puente, J.A. (2013). Static analysis of WCET in a satellite software subsystem. In C. Maiza (ed.), *13th International Workshop on Worst-Case Execution Time Analysis*, volume 30 of *OpenAccess Series in Informatics (OASIcs)*, 87–96. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.

ISO/IEC (2005). *ISO/IEC TR 24718:2005 — Guide for the use of the Ada Ravenscar Profile in high integrity systems.* ISO. Based on the University of York Technical Report YCS-2003-348 (2003).

Lovera, M. (2015). Magnetic satellite detumbling: The b-dot algorithm revisited. In *Proceedings of the American Control Conference*, 1867–1872.

Salazar, E., Alonso, A., and Garrido, J. (2014). Mixed-criticality design of a satellite software system. *IFAC Proceedings Volumes*, 47(3), 12278 – 12283. 19th IFAC World Congress.

Schwartz, J.L., Peck, M.A., and Hall, C.D. (2003). Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, 26(4), 513–522.

Sidi, M.J. (1997). *Spacecraft Dynamics and Control: A Practical Engineering Approach.* Cambridge University Press.

Silani, E. and Lovera, M. (2005). Magnetic spacecraft attitude control: a survey and some new results. *Control Engineering Practice*, 13(3), 357 – 371.

Zamorano, J. and Garrido, J. (2015). Schedulability analysis of PWM tasks for the UPMSat-2 ADCS. In J. de la Puente and T. Vardanega (eds.), *Reliable Software Technologies — Ada-Europe 2015*, volume 9111 of *Lecture Notes in Computer Science*, 85–99. Springer Berlin Heidelberg.