MODEL BASED DESIGN OF THE SENTINEL-2 ATTITUDE CONTROL SYSTEM

G. Wiedermann, S. Winkler, W. Gockel, D. Reggio, J. Levenhagen
EADS Astrium Satellites
88090 Immenstaad
Germany

Abstract

Modelling and simulation plays a major role in the development, validation and verification of a satellite's attitude control system since test flight is impossible and even proper stimulation of sensors as well as the on ground testing of the actuators' interaction with the satellite's free-free dynamics is rather expensive if not impossible, too.

After an overview of the Sentinel-2 mission and satellite and an introduction of the attitude control system this paper focuses on the modelling and simulation tools for the development, performance validation and implementation verification of the attitude control algorithms.

The main “working horse” for the attitude control algorithms development and performance validation is the AOCS Offline Simulation Environment (AOSE) which consists of models for the space environment, (e.g. gravity field, magnetic field and atmospheric density), the interaction of the space environment with the satellite leading to forces and torques, sensor and actuator models and the satellite dynamics including flexible appendages.

The core of the AOSE is the attitude control algorithms to be developed. The main purpose of the AOSE is to allow fast implementation, validation and verification of the attitude control algorithms and extensive simulation and testing against the attitude control requirements in a high fidelity simulation environment early in the development cycle. The outputs of this development step are executable and tested prototype algorithms which serve as specification for the on-board software development. Further the fundamental performance of the algorithms is already validated at this stage.

After the actual on-board software for the attitude control algorithms has been implemented the correct function of the software is verified in a similar environment called the AOCS Functional Verification Bench (AFVB) concluding the development, validation and verification of the attitude control algorithms. The further validation and verification of the on-board software is then performed on a series of test benches including real time hardware in the loop systems.

1. PROJECT OVERVIEW

1.1. The Mission

Sentinel-2 is a European polar orbit satellite system built on a constellation of two similar satellites for the provision of operational land services based on optical earth observation payload at medium spatial resolution, but over a large swath for fast revisiting as continuity and further enhancement of SPOT and Landsat.

Sentinel-2 is being built by EADS Astrium Satellites in the framework of the European Space Agency's (ESA) Global Monitoring for Environment and Security (GMES) program [1]. After successful conclusion of the phase B development with the Preliminary Design Review (PDR) end of 2008 the program is now in the detailed design phase. The launch of the first satellite is planned for 2012.

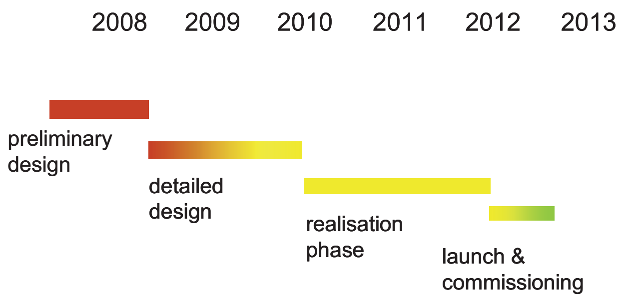

FIG 1. Overall project schedule

The two satellites will be in a polar sun synchronous orbit with 786 km altitude and 10:30 local time of the descending node. The orbital period is about 100 min. Image acquisition takes place over all land surfaces on the daylight side of the orbit.

The satellites are flying in three axis stabilised earth oriented attitude with a side-looking capability to access any point on earth within one to two days.

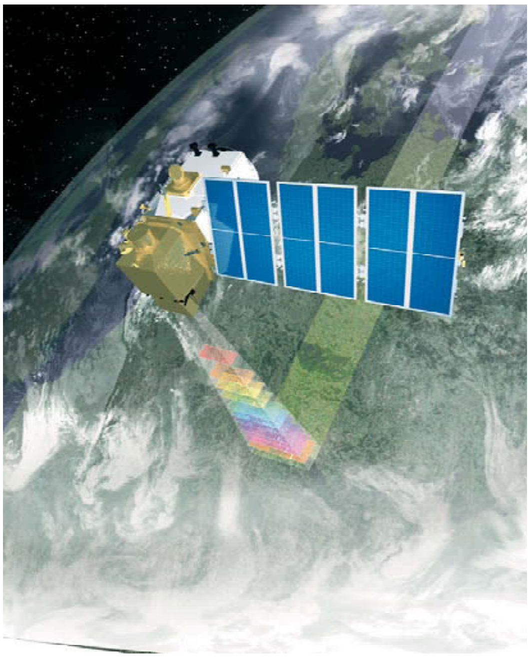

FIG 2. Sentinel-2 in operation

1.2. The Satellite

The satellite has overall dimensions of 3 x 1.7 x 2.25 m and a total mass of 1100 kg. It is based on an aluminium honeycomb structure and carries a total amount 133 kg of fuel mainly for orbit correction manoeuvres. The design is qualified for a life time of 7 years with consumables on board for 12 years. Gallium Arsenide triple junction cells on a rotating solar array with 7.5 m² size provide about 1.7 kW of electrical power.

The main instrument is the Multi-Spectral Imager (MSI) operating in 13 spectral bands from visible to short wave infrared with up to 10 m spatial resolution. The swath width is 290 km.

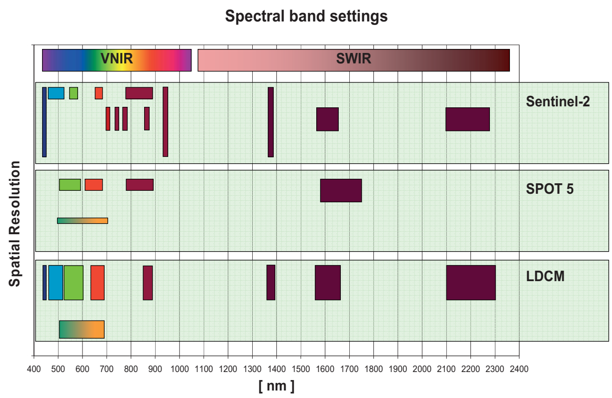

FIG 3. Multi-spectral imager spectral bands

One of the main challenges is the determination of the images geo location. Therefore three high performance star trackers (STR) and a high performance four axis gyro unit (IMU) are mounted directly on the instrument to provide input to the Gyro Stellar Estimator attitude estimation [3]. For precise position determination a high performance dual frequency GPS receiver is used.

The other sensors on board are a coarse earth and sun sensor (CESS) and two coarse three axis gyros (RMU) used for safe mode operations and three three axis magnetometers (MAG) to provide the magnetic field vector for the magnetic torquer operations in safe mode.

As actuators the satellite bus carries two times four monopropellant thrusters (THR) used for orbit correction manoeuvres and attitude control in safe mode, three magnetic torquers (MTQ) and an array of four reaction wheels (RW).

The 800 GByte image data per day are stored in a 1.7 Tbit mass memory and transmitted to ground using an X-band downlink system.

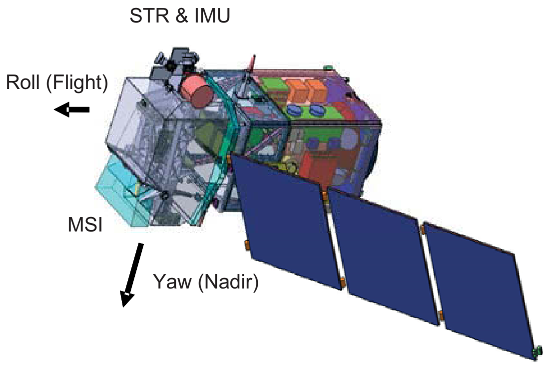

FIG 4. The Sentinel-2 satellite

1.3. The Attitude Control System

1.3.1. Modes, Sensors and Actuators

The task of the attitude control system is to autonomously acquire and hold the required satellite attitude within specified attitude error limits. This task requires a closed loop control system. In contrary orbit control is not performed autonomously but by ground command.

The attitude control system has three main modes with several submodes each:

- **Initial acquisition and safe mode (ASM)**

  This mode is entered after separation from the launcher and in case of a failure. Its purpose is to damp out the satellite rates and acquire a coarse earth pointing attitude from any initial condition to ensure safe thermal and power conditions for the satellite. Since this mode is also used as safe mode in case of a failure the sensors and actuators shall be independent from the sensors and actuators used in normal observation mode.

When the mode is entered the satellite rates measured with the coarse gyro are damped (submode ASM RD). As soon as the residual rates are sufficiently small the satellite acquires an earth pointing attitude using the earth vector measured by the coarse earth and sun sensor in the submode ASM EA. Following earth acquisition the satellite is rotated around the yaw axis into the proper flight direction.

The actuators used in ASM are the thrusters for fast rate damping and acquisition supported by magnetic torquers to minimise fuel consumption in steady state. Magnetometers are used to measure the magnetic field to compute the proper magnetic moments to be commanded to the magnetic torquers.

- **Normal mode (NOM)**

  This is the mode for the nominal operations of the instrument. When entering from ASM the first submode is the attitude hold mode (NOM AH), a robust mode which moves the satellite from the coarse attitude during ASM with attitude errors up to 30° into a much more accurate earth pointing attitude with pointing errors in the millirad range before switching fine pointing mode (NOM FP) for observation.

  To allow the instrument to look sideward the satellite can be rotated about 20° around the roll axis (extended fine pointing - NOM EFP) using the slew (NOM SL) and back slew (NOM BSL) submodes.

  The sensors used in NOM for attitude estimation are the three high performance star trackers and the high performance gyros, the actuators are the four reaction wheels. The magnetic torquers are used to dump the angular momentum stored in the reaction wheels and avoid saturation of the wheels.

- **Orbit control mode (OCM)**

  The orbit control mode is used to perform orbit correction manoeuvres. The first submode is a slew mode (OCM SL) to rotate the satellite around the yaw axis and orient the thruster in the right direction. This is followed by a stabilisation period (OCM STAB) before the thrusters are fired (delta-v submode - OCM DV). Finally the satellite is rotated back in its nominal flight orientation.

  With exception of the OCM DV submode the sensors and actuators used are similar to the normal mode. When the thrusters are fired for the orbit control manoeuvre the attitude is controlled by off-modulation of the thrusters.

All processing needed for attitude control is done in the on-board computer (OBC) which is connected to the remote interface unit (RIU) which establishes the analogue and digital interfaces to the sensors and actuators.

An overview of all modes and submodes is given in FIG 5 while FIG 6 shows the hardware used for the attitude control system.

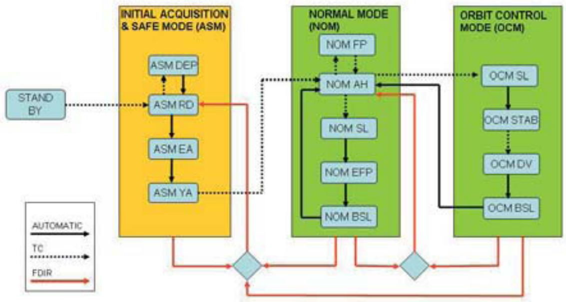

FIG 5. Mode diagram

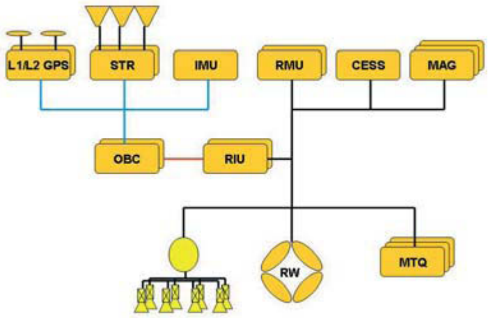

FIG 6. Attitude control hardware

1.3.2. Software

The attitude control application is part of the central software (CSW) implemented in the on-board computer. We distinguish the attitude control algorithms (inner box in FIG 7) and the data handling part (outer part of FIG 7).

The control algorithms contain the functional part of the attitude control software comprising:

- **Sensor processing**
  Takes the raw data from the sensors and transforms it into a format and coordinate frame needed for further processing.
- **State estimation**
  Uses sensor date to estimate e.g. attitude, position etc. of the satellite.
- **Controller**
  Computes the control torques according to the control laws.
- **Guidance**
  Computes the reference attitude.
- **Actuator commanding**
  Transforms the commanded control torques into actuator specific commands in raw format (thruster on-time, reaction wheel torque, magnetic moment for magnetic torquer)

- **Status monitoring and FDIR**
  Monitors the status of the attitude control system and performs failure detection, isolation and recovery as far as required at this level
- **Mode management**
  Switch main- and submodes of the attitude control system

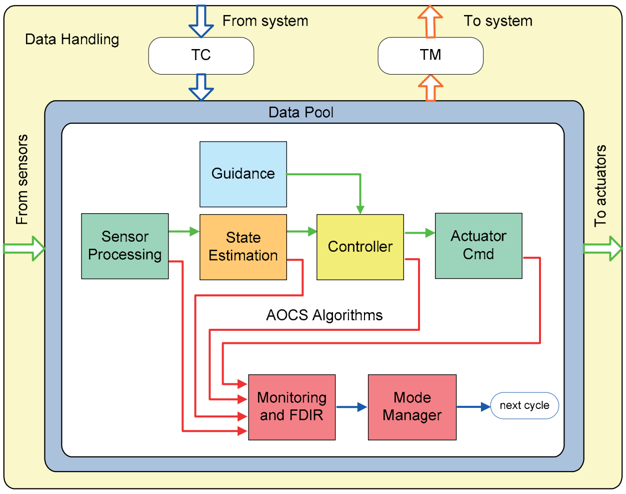

FIG 7. Attitude control software architecture

The data handling part is responsible for the acquisition of sensor data and the transmission of commands to the actuators as well as the handling of telecommands and telemetry packet. Although the data handling part is an important part of the attitude control application it is rather generic in nature. Therefore we will only focus on the control algorithms part where model based design plays an important role.

The interface between the data handling part and the attitude control algorithms is a so called data pool, a shared memory area where the data is exchanged. Therefore the attitude control algorithms have a clear interface with no direct access to hardware or operating system functions. This is important to allow easy cut out of the control algorithms part for stand alone development and testing.

1.3.3. Development, Validation and Verification Steps

The preceding chapters gave an overview of actual Sentinel-2 attitude control system and therefore presented the outcome of the development process. But what are the steps to get there? A short overview:

1) Analysis and understanding of requirements
2) Break down of top level requirements to subsystem level
3) Definition of modes, actuators and sensors
4) Definition of control algorithms functions
5) Development of control algorithms
6) Performance validation of control algorithms
7) Specification of control algorithms for the software developer
8) Implementation of control algorithms software
9) Performance validation and verification of control algorithms software
10) Integration with central software
11) Performance validation and verification of attitude control system in several steps with increasing complexity (all simulation - real time simulation - hardware in the loop)

In the following we will concentrate on steps 5 to 11 where model based design plays an important role.

2. MODEL BASED DESIGN

2.1. Understanding of Model Based Design

Model based design to us means to use simulation and simulation models right from the start of the development until the final verification. This also allows continuous testing and validation right from the start in a realistic environment.

Although the level of detail and the proximity to reality increases throughout the development, validation and verification life cycle, the core functionality of the simulation models is inherited from one step to the next. Only additional layers are added to cope with increasing demands.

In the next chapters let us introduce several of the simulation environments used for the Sentinel-2 attitude control system.

2.2. AOCS Offline Simulation Environment

The first important simulation environment is the AOCS Offline Simulation Environment (AOSE) used to develop the attitude control algorithms and perform the first performance validation to show that the attitude control system is able to cope with the requirements.

Therefore the simulation environment has to provide all relevant effects to validate the performance of the attitude control algorithms. The main elements as sketched in FIG 8 are:

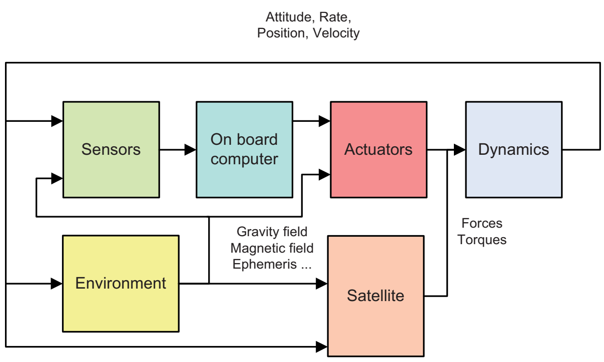

FIG 8. AOSE top level architecture

- **Environment Model**
  This model computes the environmental quantities which are only dependent on the satellite position. This includes

- time
- earth rotation parameters
- sun and moon ephemeris
- earth gravitational field (JGM3)
- sun and moon attraction
- earth magnetic field (IGRF 10)
- earth atmospheric density (MSIS 86)
- earth atmospheric velocity
- sun flux
- conical eclipse

- **Satellite Model**
  This model covers the interaction of the satellite with the environment. For example:
  - Gravity force and torque
  - Torque due to residual magnetic moment
  - Aerodynamic force and torque
  - Solar pressure force and torque
  - Solar, earth Albedo and earth infrared loads on defined faces

  The satellite model includes a geometrical model of the satellite which can have degrees of freedom (e.g. rotation of a solar array) and is used for the computation of the last three quantities.

- **Dynamics Model**
  The dynamics model performs the integration of the satellite's equations of motion considering degrees of freedom and flexible appendages.

- **Sensors and Actuators**
  High fidelity models of the sensors and actuators used.

- **On-board Computer**
  The on-board computer model contains the interface blocks between hardware and software (sensor data acquisition, actuator drivers) which resembles the RIU and the item to be developed and tested - the prototype attitude control algorithms.

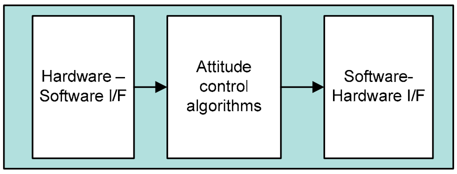

FIG 9. On-board computer model

The AOSE is implemented as Simulink model which provides an easy to use user interface and excellent pre- and post-processing capabilities using the Matlab environment [2]. The prototype attitude control algorithms which will serve as specification for the implementation of the attitude control algorithms on-board software are implemented using Embedded Matlab for the functional algorithms and masked Simulink subsystems for the architecture.

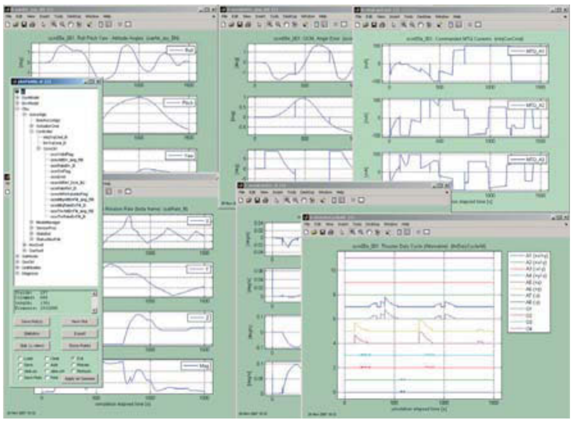

FIG 10. AOSE output

AOSE	
Purpose	Development and performance validation of attitude control algorithms
Implementation	Simulink
Equipments	Core functional model
OBC	Functional simulation of hardware - software interfaces
Software	Prototype attitude control algorithms

TAB 1. AOCS offline simulator environment summary

2.2.1. AOCS Functional Verification Bench

The AOCS Functional Verification Bench (AFVB) is similar to the AOSE and uses the same models. The difference is in the item under test. While the AOSE contains the prototype attitude control algorithms the AFVB contains the attitude control algorithms on board software.

The AFVB is used to verify the implementation of the attitude control algorithms and validate the performance with the actual on-board implementation.

The AFVB can be build in two ways. One way is to integrate the attitude control algorithms software with a proper wrapper as an S-function into the Simulink AOSE Simulink model thus effectively enlarging the already existing AOSE, the other way would be to use the C / C++ version of the environment, satellite, dynamics and equipment models and integrate the attitude control software there.

AFVB	
Purpose	Performance validation and verification of attitude control algorithms on-board software
Implementation	Simulink or C / C++
Equipments	Core functional model
OBC	Functional simulation of hardware - software interfaces
Software	Attitude control algorithms on-board software

TAB 2. AOCS offline simulator environment summary

2.2.2. Software Verification Facility

Starting with the Software Verification Facility (SFV) the scope is no longer the attitude control system but the validation and verification of the whole on-board software running inside a high fidelity on-board computer simulator.

SVF	
Purpose	On-board software validation and verification
Implementation	C / C++
Equipments	Core functional model + complementary model + model interface layer
OBC	OBC emulator
Software	On-board software

TAB 3. Software verification facility summary

2.2.3. Real Time Test Benches

For further validation and verification including closed loop attitude control system tests a series of real time test benches with the OBC as hardware in the loop is used. Other equipments can be simulated or included as hardware in the loop. A TM / TC front end can be used to test operational procedures.

Even if the scope is now at its maximum perimeter, the models used are still the same as in the AOSE.

Real time test benches	
Purpose	System validation and verification
Implementation	C / C++, real time system
Equipments	Core functional model + complementary model + model interface layer Hardware in the loop
OBC	OBC as hardware in the loop
Software	On-board software

TAB 4. Real time test benches summary

2.3. Tools and Libraries

Many of the tools and models used for the simulation environments are reused for different projects. Therefore efficient project work calls for heavy use of libraries and templates to ensure reuse of existing tools and models and high quality due to continuous maintenance.

A continuous effort is going on in Astrium Satellite's AOCS division to establish and maintain these libraries. Cross national working groups have been established which also work together with other divisions to optimise the attitude control system development, validation and verification process.

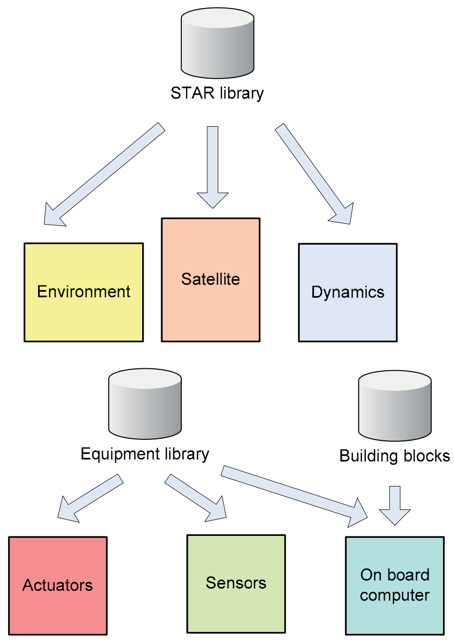

FIG 11. Libraries

2.3.1. AOSE Template

The top level architecture and the contents of the environment, satellite and dynamics models are very similar for many projects, especially in the realm of earth observation, and can therefore be standardised. After use of the AOSE approach in several projects a formal Simulink template is under finalisation which fits the needs of many future projects.

2.3.2. STAR Library

The STAR library contains basic functions needed for analysis and simulation of the attitude control system. For example most of the contents of the environment, satellite and dynamics models come from the STAR library.

Many elements of the library are based on C-code which is interfaced as Simulink S-function and Matlab mex - function for the use in the Matlab / Simulink environment. This approach allows using the same functions also in the C environment of other simulators and test benches later in the validation and verification process.

2.3.3. Building Blocks Library

In the attitude control algorithms often the same basic functional elements are used in different arrangements. Therefore a building blocks library is under development to standardise and maintain the basic building blocks (e.g. pulse width modulator, orbit propagator, magnetic field model, magnetic torquer commanding …).

On the next higher functional level these building blocks are combined into AOCS functions which can stretch over the whole functional chain from sensor processing to actuator commanding (see FIG 7). For example the reaction wheel array management function contains the processing of the reaction wheel rate sensors, the computation of the reaction wheel angular momenti, and the control of the array's null space and the commanding and friction estimation of the individual wheels.

Although the development is still ongoing, several readily available and validated building blocks and functions are already available to be used in the Sentinel-2 project. The building blocks are implemented using MathWorks' Embedded Matlab language and thus can easy be integrated into the AOSE to build up the attitude control algorithms. Further Embedded Matlab is easy to read for documentation of the attitude control algorithms.

For each AOCS function there will be a counterpart in the on-board software which is readily available and verified thus saving implementation and verification effort.

2.3.4. Equipment Models Library

For realistic simulations high fidelity equipment models for sensors and actuators are needed. However, the requirements for functionality and interfaces increase throughout the development, validation and verification process.

The AOSE and AFVB are focused on the attitude control algorithms. Therefore the equipment model has to contain the functionality and interfaces needed for this task. This is achieved by the core functional model.

On later test benches the focus is enlarged to cover the whole on-board software and system aspects. Therefore the equipment models have to be adapted also to contain more functionality (e.g. star tracker image dump) and provide a representative interface covering also operational aspects. This is done by adding the complementary model.

Finally when it comes to hardware in the loop (e.g. the OBC) a layer is needed to for the communication of the models with the hardware. This job is performed by the system interface layer

Since the core functional model contains mostly the physics of the sensor or actuator is obviously the less hardware specific part of the model and therefore well suited to be put in a library which is currently under development. In many cases the core functional model can be reused without or only minor changes (e.g. the output format). The complementary model and the system interface layer are specific for a certain piece of equipment and are likely to change for different suppliers or different models from one supplier.

Another part of the equipment library are the functional models for the OBC hardware interfaces, e.g. A/D, D/A conversion, magnetic torquer driver, thruster driver etc.

The equipment models are implemented using Embedded Matlab and transferred to C where necessary.

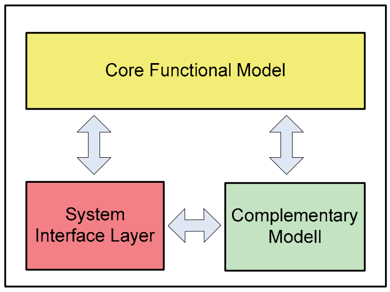

FIG 12. Equipment model structure

2.3.5. Executable Specification

After development and validation of the attitude control algorithms using the AOSE the attitude control algorithms on-board software has to be produced. Therefore a specification has to be given to the software developer.

In the case of Sentinel-2 the main part oft the specification is a written document produced using DOORS which can easily be reviewed and used as reference. However, the prototype control algorithms code, which has already been tested and validated, is directly used for the specification thus avoiding errors, misunderstandings and ambiguities. Since the prototype code is written using Embedded Matlab it is easy readable and can be regarded as a kind of pseudo code. The architecture of the algorithms is represented by Simulink blocks.

On top of that the Simulink blocks containing the code are given to the software supplier together with input and reference output date as an executable reference.

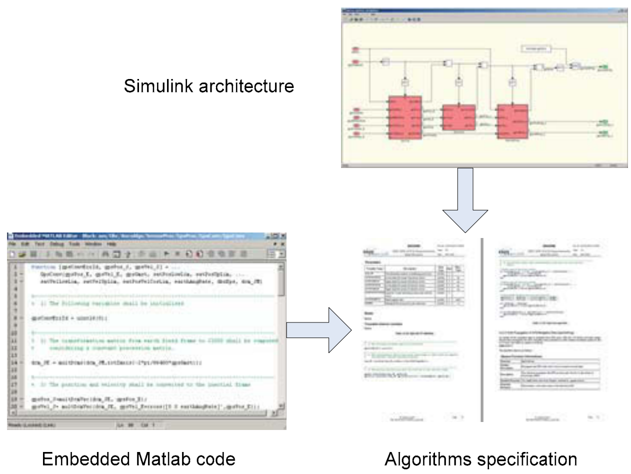

FIG 13. Composition of the attitude control algorithms specification.

2.4. Development Steps

After the introduction of the various simulation environments and the tools and libraries used let's now look at the development, validation and verification steps starting with control algorithms development again and see how they are performed with the model based design approach.

The AOSE is the main working horse for the development of the attitude control algorithms where the starting point is the AOSE template with an empty attitude control algorithms part.

After parameterisation of the AOSE the implementation of the attitude control algorithms can be started. Typically this is done mode by mode focussing on the functional aspects first followed by monitoring, FDIR and mode management. Since the AOSE is already ready to run continuous testing s possible from the start. As soon as the first mode is implemented the performance can be validated against the requirements.

In the Sentinel-2 project at the end of the preliminary design phase all modes were implemented and their performance was validated. Together with the design description, the stability analysis and the tuning of the attitude control algorithm parameters the simulations performed on the AOSE were a major for the preliminary design review. The next step was the design and the implementation of the FDIR and mode management part.

As soon as all algorithms are implemented and the performance is validated it is time to produce the attitude control algorithms specification for the software developer. Typically this will be done in several steps starting with the safe mode algorithms. Once the attitude control algorithm on-board software is written and the open loop acceptance tests, for which the AOSE serves as reference, are passed the software is implemented into the AFVB for verification and final validation.

In parallel validation and verification activities for the on-board software are performed on the SVF before the final validation and verification campaign in the real time environment with hardware in the loop takes place.

3. CONCLUSIONS

The model based design approach for the attitude control algorithms allows continuous testing and validation in a high fidelity simulation environment right from the start of the development thus avoiding design flaws due to oversimplification.

Tested and validated executable prototype code can be used as specification for the on-board software implementation which reduces the danger of incomplete, ambiguous or wrong specifications and misinterpretations which would require costly corrections in a later stage of the software development.

The cost of the development is further reduced by proper modularisation enabling heavy use of libraries for both simulation environments and the algorithms and software under development. The same models are used throughout the whole development, validation and verification chain.

Although today the transition from prototype attitude control algorithms to on-board software as well as the transformation of models for different platform is done manually the model based approach is the basis for automatic code generation further reducing the software development effort.

4. REFERENCES

[1] European Space Agency, ESA's Sentinel satellites - Sentinel-2, http://www.esa.int/esaLP/SEMM4T4KXMF_LPgmes_0.html

[1] The MathWorks, Homepage, http://www.mathworks.de/

[2] S. Winkler, G. Wiedermann, W. Gockel. High-Accuracy On-Board Attitude Estimation for the GMES Sentinel-2 Satellite: Concept, Design, and First Results. Proceedings of the AIAA Guidance, Navigation, and Control Conference 2008 (Honolulu, HI, August 18-21, 2008), AIAA-2008-7482