AOCS Software

Starting from the flight experience obtained with the Danish Ørsted satellite, Terma has developed a generic AOCS software system which has been employed as basis for the design of the Herschel and Planck missions and influenced the AOCS software for the Solar Orbiter and Sentinel-6 missions.

Implementing the AOCS system for the Ørsted satellite clearly demonstrated that a strict modeling approach and a clear differentiation between algorithm development and software engineering had to be respected. In addition, it was found that risks primarily arise from dynamic control flows interacting in ways that are neither well understood nor properly modelled.

Follow up design studies for ESA demonstrated the feasibility of introducing a generic AOCS software architecture suitable for several classes of spacecraft.

These results – combined with an engineering tradition for highly reliable embedded software – led to the implementation of the AOCS software for the Herschel/Planck missions and paved for Terma’s development of the AOCS software for the Solar Orbiter and Sentinel-6 missions.

Reusable AOCS software concept

The concept of a re-useable AOCS software kernel is clearly demonstrated on the Herschel/Planck missions where Herschel is 3-axis stabilized and Planck is spin stabilized. It is based on a layered structure comprising:

- A system interface layer that provides the operational interface to the AOCS application software, handling service requests in the form of telecommands, and generating service reports in the form of telemetry packets.
- A rule layer that implements command execution and autonomy with graceful degradation in response to subsystem failure.
- A process layer that implements the attitude and orbit control algorithms.
AOCS Software

- A device interface layer that manages and encapsulates the commanding and reading of devices.

Both the nominal modes and survival modes of the two spacecraft are being implemented with a common top-level architecture. To accommodate differences between spacecrafts, the functional behavior of the individual components is modified whilst the component interfaces are kept identical for all configurations (the design-by-contract paradigm).

Graceful degradation – calling for dynamic reconfiguration of sensors and actuators – is implemented through a rule-based reconfiguration logic that defines transitions to a degraded configuration for failure recovery.

And when it comes to operations, it should be noted that ESA’s Packet Utilization Standard are fully supported by the design.

Design features
The guiding principles for the design have been:

- System level mode transitions - and resulting reconfigurations - are centralized to maintain consistency and to base autonomous transitions on a system level perspective.
- Failure detection is distributed across the system i.e. accommodated as close to the originating source as possible.
- Failure isolation and recovery is centralized and takes place at system level.
- All external interfaces are encapsulated in ‘wrapper’ interface components to mitigate consequences of external failures and to accommodate subsequent modifications of interfaces.
- General control engineering is decoupled from vendor specific actuator and sensor characteristics, operating on calibrated engineering values.

All interfaces are designed to capture and reflect the specific temporal characteristics and to provide the necessary safety barriers. This includes the System Interface that supports telecommand flow through polling of the command queue, general routines for generation of event reports, and depositing of application software parameters for potential inclusion in housekeeping and diagnostics reports.

All software components are parameterized over the contents of an onboard database. This supports reading and modifying assigned values for the entire set of parameters for the AOCS software.

The resulting generalized architecture includes the following high level components:

- Attitude determinator derives an estimate of the current spacecraft attitude based on readings acquired from the various sensors active in the given configuration.
- Controller derives the required torque and/or force required to acquire a given set-point or delta-V. It subsequently commands the various actuators activated in the given configuration.
- Supervisor is in control of the overall software configuration and responsible for device reconfiguration and mode transitions, reacting either upon explicit request through a telecommand, or autonomously in response to on-board events.
- Failure Detection detects single point failures distributed across all components. Events are reported to Failure Detection which isolates the error and requests a response at system level.
- High-Level Actuator Management which includes a Reaction Control System Manager handling torque-force distribution and local thrusters fault detection, and a Reaction Wheels Manager implementing torque distribution, local fault detection, and momentum unloading.
- High-Level Sensor Management which encompasses a Gyroscopes Manager, a Star Tracker Manager, a Sun Sensor Manager, and a Coarse Rate Sensor Manager handling the various AOCS sensors.