University of Constance
Department of Computer Science

AOCS FRAMEWORK - AOCS PROTOTYPE DEFINITION

Abstract

This document was written as part of the study "Design and Prototyping of a Software Framework for the AOCS" done under contract Estec/13776/99/NL/MV for ESA-Estec. The purpose of the study is the development of a software framework for the Attitude and Orbit Control Subsystem (AOCS) of a satellite. The framework was tested by using it to generate the software for a prototype AOCS. This document defines the characteristics of the AOCS prototype.

Written By:	A. Pasetti
Date:	30 April 2002
Issue:	1.1
Reference:	SWE / 99 / AOCS / 020

TABLE OF CONTENTS

1 REFERENCES

RD1 E. Gamma *et al.* (1995), *Design Patterns – Elements of Reusable Object Oriented Software*, Reading, Massachusetts: Addison-Wesley

RD2 A. Pasetti (2000), *AOCS Framework – Concept Level Description*, AOCS Framework Document ref. SWE/99/AOCS/004

RD3 Deleted

RD4 Deleted

RD5 A. Pasetti (2000), *Inter-Component Communication Framelet* AOCS Framework Document ref. SWE/99/AOCS/005

RD6 A. Pasetti (2000), *Object Monitoring Framelet*, AOCS Framework Document ref. SWE/99/AOCS/008

RD7 A. Pasetti (2000), *Data Processing Framelet*, AOCS Framework Document ref. SWE/99/AOCS/006

RD8 A. Pasetti (2000), *AOCS Unit Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/017

RD9 A. Pasetti (2000), *Reconfiguration Management Framelet* AOCS Framework Document ref. SWE/99/AOCS/015

RD10 A. Pasetti (2000), *Operational Mode Management Framelet* AOCS Framework Document ref. SWE/99/AOCS/009

RD11 T. Brown, A. Pasetti (2000), *Manoeuvre Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/012

RD12 A. Pasetti (2000), *Failure Detection Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/010

RD13 A. Pasetti (2000), *System Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/021

RD14 A. Pasetti (2000), *Failure Recovery Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/011

RD15 A. Pasetti (2000), *Telemetry Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/003

RD16 A. Pasetti (2000), *Telecommand Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/014

RD17 A. Pasetti, T. Brown (2000), *Controller Management Framelet*, AOCS Framework Document ref. SWE/99/AOCS/016

RD18 A. Pasetti (2000), *AOCS Prototype Definition*, AOCS Framework Document ref. SWE/99/AOCS/020

RD19 *MACS Bus Handbook*

2 ACRONYMS

AAD	Attitude Anomaly Detection
AOCS	Attitude and Orbit Control Subsystem
AST	Autonomous Star Tracker
CSS	Coarse Sun Sensor
ES	Earth Sensor
FDIR	Failure Detection, Isolation and Recovery
FPM	Fine Pointing Mode
FSS	Fine Sun Sensor
GYR	Gyroscope
KF	Kalman Filter
IAM	Initial Acquisition Mode
MIMO	Multi-Input-Multi-Output
NM	Normal Mode
NTT	Non-Time-Tagged
OBDH	On-Board Data Handling system (aka as OBDS)
OCM	Orbit Control Mode
OO	Object-Oriented
PD	Proportional-Derivative controller
PI	Proportional-Integral controller
PID	Proportional-Integral-Derivative controller
RRM	Rate Reaction Mode
RTOS	Real-Time Operating System
RW	Reaction Wheel
SAS	Sun Attitude Sensor
SBM	Stand-By Mode
SISO	Single-Input-Sinle-Output
SPS	Sun Presence Sensor
STR	Star Tracker
SLM	Slewing Mode
SM	Safe Mode
TC	Telecommand
THU	Thruster
TM	Telemetry
TT	Time-Tagged

3 INTRODUCTION

This document describes the *AOCS prototype* used to test the *AOCS prototype framework*. The AOCS prototype is a simplified AOCS software which is implemented using the constructs offered by the AOCS prototype framework. The AOCS prototype thus serves as a test bed for the AOCS prototype framework.

This document should not be regarded as a full set of user requirements for an AOCS software. Its aim is only to give as much information as is needed to instantiate the AOCS prototype software from the AOCS prototype framework. In particular, this document does *not* contain sufficient information to set up a “real-world model” for a closed-loop simulation of the prototype AOCS.

The AOCS prototype is not intended to be representative of any real AOCS. Its interest lies simply in the extent to which it allows the functionalities of the AOCS prototype framework to be exercised and the constructs exported by it to be utilized.

3.1 Context

The context for the definition of the prototype framework is the architectural design of the full framework. This is presented at system concept definition level in RD2 and at framelet concept and at framelet architectural definition level in RD5 to RD17. The prototype framework is defined in RD18.

3.2 AOCS and Spacecraft Parameter Values

This document gives the spacecraft and AOCS parameter values as symbolic constants. Their numerical values can be found by inspecting the relevant `include` files. In particular, the parameters that are needed by the AOCS software are defined in file `AocsConfiguration.h`. Thus, for instance, the thruster torque level is designated in this document as T_THU. A corresponding `#define` variable is declared together with its numerical value in the `AocsConfiguration.h` file.

Some of the parameters used in this document are only relevant if a real-world simulation is coupled to the AOCS. If the AOCS prototype is tested in open-loop mode (no real-world simulation) their value becomes irrelevant.

4 MISSION SCENARIO

The mission scenario assumed by the prototype AOCS is a 3-axis stabilized satellite. The nominal attitude of the spacecraft is as follows:

- Spacecraft Z axis sun-pointing
- No angular rate around the spacecraft Z axis
- No pointing requirements on the spacecraft Z axis

4.1 Spacecraft Modes

The prototype AOCS only covers the part of the spacecraft mission. More specifically, it covers the following operational modes:

- *Coarse Sun Pointing (CSP)*

  The spacecraft maintains its nominal attitude with a low accuracy, namely pointing errors around the X- and Y-axes of CSP_X_ATT_ERR and CSP_Y_ATT_ERR radians and Z-axis angular rate below CSP_Z_RATE_ERR rad / sec.

- *Fine Sun Pointing (FSP)*

  The spacecraft maintains its nominal attitude with a high accuracy, namely pointing errors around the X- and Y-axes of FSP_X_ATT_ERR and FSP_Y_ATT_ERR radians and Z-axis angular rate below FSP_Z_RATE_ERR rad / sec.

4.2 Spacecraft Characteristics

The AOCS sees the spacecraft as a rigid body with a diagonal inertia matrix. The moments of inertia around the three spacecraft axes are: SC_INERTIA_X, SC_INERTIA_Y and SC_INERTIA_Z.

4.3 Orbit Control

Orbit control is not part of the prototype AOCS mission scenario.

5 AOCS SENSORS

The prototype AOCS software controls the following complement of attitude sensors:

- 3 Sun acquisition sensors (prime + redundant)
- 1 Fine sun sensor (prime + redundant)
- 1 Gyro (prime + redundant)

5.1 Sun Acquisition Sensors (SAS)

The prime SAS's are designated as SAS_1_A to SAS_3_A. The redundant SAS's are designated as SAS_1_B to SAS_3_B

The boresight of SAS_1_A is nominally aligned with the spacecraft Z axis. The nominal boresight of the other two SAS`s is identified by the following unitary vectors in the spacecraft reference frame:

- SAS_2_A : (SAS_2_BS_X, SAS_2_BS_Y, SAS_2_BS_Z)
- SAS_3_A: (SAS_3_BS_X, SAS_3_BS_Y, SAS_3_BS_Z)

Prime and redundant SAS's have the same nominal attitudes.

Each SAS generates two measurements corresponding to the sun position along the sensor's X and Y axes.

The prototype AOCS software applies a bias and scaling factor correction to the raw measurements from each SAS sensor. In other words, if `rawSasX` and `rawSasY` are the raw SAS outputs expressed in engineering units, the prototype AOCS software uses the following formula to derive the sun angles in radians:

```
sasX = biasSasX + scalingSasX * rawSasX
sasY = biasSasY + scalingSasY * rawSasY
```

The bias and scaling correction factors for each SAS are:

SAS_i_k :	BIAS_SAS_i_k_X, SCALING_SAS_i_k_X,
	BIAS_SAS_i_k_Y, SCALING_SAS_i_k_Y

Where i ranges from 1 to 3 for SAS_1 to SAS_3 and k is either 'A' or 'B' for the prime and redundant branch.

5.2 Fine Acquisition Sensor (FSS)

The prime and redundant FSS's are designated as FSS_A and FSS_B. Their boresight is parallel and nominally aligned with the spacecraft Z axis.

Each FSS generates two measurements corresponding to the sun position along the sensor's X and Y axes.

The prototype AOCS software applies a bias and scaling factor correction to the raw measurements from an FSS. In other words, if `rawFssX` and `rawFssY` are the raw FSS outputs expressed in engineering units, the prototype AOCS software uses the following formula to derive the sun angles in radians:

```
fssX = biasFssX + scalingFssX * rawFssX
fssY = biasFssY + scalingFssY * rawFssY
```

The bias and scaling correction factors for the prime FSS are:

FSS_A : BIAS_FSS_A_X, SCALING_FSS_A_X, BIAS_FSS_A_Y, SCALING_FSS_A_Y,

The bias and scaling factors for the redundant FSS have similar names.

5.3 Gyro (GYR)

The prime and redundant GYR's are designated as GYR_A and GYR_B. Their boresight is parallel and nominally aligned with the spacecraft Z axis.

Each GYR generates a single measurements corresponding to the angular rate around its sensing axis.

The prototype AOCS software applies a bias and scaling factor correction to the raw measurement from a GYR. In other words, if `rawGyrZ` is the raw GYR output expressed in engineering units, the prototype AOCS software uses the following formula to derive the angular rate radians / sec:

```
gyrZ = biasGyrZ + scalingGyrZ * rawGyrZ
```

The bias and scaling correction factors for the prime GYR are:

GYR_A: BIAS_GYR_A, SCALING_GYR_A

The bias and scaling factors for the redundant GYR have similar names.

6 AOCS ACTUATORS

The prototype AOCS software controls the following complement of attitude actuators:

- 4 Reactions wheels (internally redundant)
- 6 Thrusters (prime + redundant)

Both reaction wheels and thrusters are arranged so as to be able to produce torques around all three spacecraft axes.

6.1 Reaction Wheels (RW)

The four reaction wheels are designated as RW_1 to RW_4.

The reaction wheels axes expressed in the spacecraft coordinate frame are:

- RW_1 : (0, 0, 1)
- RW_2 : (0, 1, 0)
- RW_3 : (1, 0, 0)
- RW_4 : (0.58, 0.58, 0,58)

From each reaction wheel a single measurement can be acquired representing the reaction wheel speed. The prototype AOCS software does not apply any correction to the speed measurements from the reaction wheels.

Each reaction wheel accepts a single command representing the torque request for the wheel.

6.2 Thrusters (THU)

The six thrusters of the prime branch are designated as THU_1_A to THU_6_A. The corresponding thrusters on the redundant branch are designated as THU_1_B to THU_6_B.

The nominal torque generated by thrusters in the prime branch is T_A. The nominal torque generated by thrusters in the prime branch is T_B. The torque generated around each axis by each thruster in the prime branch is as shown in the table below:

Thruster	SC X Axis	SC Y Axis	SC Z Axis
THU_1	T_A	0	0
THU_2	- T_A	0	0
THU_3	0	T_A	0
THU_4	0	- T_A	0

THU_5	0	0	T_A
THU_6	0	0	- T_A

The torques generated by the redundant branch are the same except that T_A is replaced by T_B.

Each thruster can receive two commands: the *on-time* and the *delay-time*. The prototype AOCS always commands zero delay-times. The commanded on-time for the prime branch is obtained as:

```
CommandsOnTime = T_A * controlPeriod / CommandedTorque
```

Where `commandedTorque` is the commanded torque and `controlPeriod` is the period with which the thrusters are commanded (ie. The control cycle). The formula for the redundant branch can be obtained by replacing T_A with T_B.

The control period is designated by THU_PERIOD.

In the AOCS prototype the thruster period is constant for all modes in which thrusters are used.

In the AOCS prototype the nominal torques generated by the two branches are assumed identical so that T_A=T_B. This common thruster torque is designated by T_THU.

7 MACS INTERFACE

The AOCS prototype assumes that communication between the AOCS computer and the external units is through a MACS bus implementing the MACS TC protocol (see RD19).

7.1 MACS Units Address

The units assumed by the AOCS prototype are listed in the table below together with their (symbolic) MACS bus address.

MACS Unit	MACS Bus Address
AOCS Computer	MACS_ACE
Prime Fine Sun Sensor (FSS A)	MACS_FSS_A
Redundant Fine Sun Sensor (FSS B)	MACS_FSS_B
Reaction Wheel 1 (RW1)	MACS_RW
Reaction Wheel 2 (RW2)	MACS_RW+1
Reaction Wheel 3 (RW3)	MACS_RW+2
Reaction Wheel 4 (RW4)	MACS_RW+3
Prime Sun Acquisition and Propulsion Electronics (SAP) (Controlling 3 Sun Acquisition Sensors and 6 Thrusters)	MACS_SAP_A
Redundant Sun Acquisition and Propulsion Electronics (SAP) (Controlling 3 Sun Acquisition Sensors and 6 Thrusters)	MACS_SAP_B
Prime Z-axis Gyro (GYR A)	MACS_GYR_A
Prime Z-axis Gyro (GYR A)	MACS_GYR_B

The instructions understood by each class of units are defined in the following subsections.

7.2 Fine Sun Sensor (FSS)

The MACS instructions understood by the FSS are listed in the table below.

Extension	Command	Subaddress

Description		
MACS_INSTR	MACS_RC	1
Initialize the FSS MACS bus head		
MACS_INSTR	MACS_TI	FSS_SA_X / FSS_SA_Y
Acquire the FSS X / FSS Y read-out		

The format of the FSS X and FSS Y read-out's is as follows:

Bit Fields	Value
Bits 0-12	Raw sun angle in rad with quantization level of: FSS_Q
Bits 13-14	Unused – always zero
Bit 15	Sign bit: readout is positive if bit is equal to 1

7.3 Reaction Wheel (RW)

The MACS instructions understood by each reaction wheel are listed in the table below.

Extension	Command	Subaddress
Description		
MACS_INSTR	MACS_RC	1
Initialize the RW MACS bus head		
MACS_INSTR	MACS_TI	RW_SA_SPD
Acquire the wheel speed measurement		
MACS_INSTR	MACS_RD	RW_SA_TOR
Receive the torque demand for the reaction wheel		

The format of the reaction wheel speed read-out is as follows:

Bit Fields	Value
Bits 0-7	Raw speed measurement in rad/sec with quantization level of: RW_SPD_Q

	RW_SPD_Q
Bits 8-14	Unused – always zero
Bit 15	Sign bit: readout is positive if bit is equal to 1

The format of the reaction wheel torque command is as follows:

Bit Fields	Value
Bits 0-7	Raw torque request with quantization level of: RW_TOR_Q
Bits 8-14	Unused – always zero
Bit 15	Sign bit: readout is positive if bit is equal to 1

7.4 Gyro (GYR)

The MACS instructions understood by the gyro are listed in the table below.

Extension	Command	Subaddress
Description		
MACS_INSTR	MACS_RC	1
Initialize the GYR MACS bus head		
MACS_INSTR	MACS_TI	GYR_SA_RAT
Acquire the gyro rate measurement		

The format of the gyro rate read-out is as follows:

Bit Fields	Value
Bits 0-7	Raw rate measurement in rad/sec with quantization level of: GYR_RAT_Q
Bits 8-14	Unused – always zero
Bit 15	Sign bit: readout is positive if bit is equal to 1

7.5 Sun Acquisition and Propulsion Electronics (SAP)

The MACS instructions understood by the SAP unit are listed in the table below.

Extension	Command	Subaddress
Description		
MACS_INSTR	MACS_RC	1
Initialize the SAP MACS bus head		
MACS_INSTR	MACS_TI	SAS_SA_X_1 … SAS_SA_X_3
Acquire the X channel read-outs for SAS 1 to SAS 3		
MACS_INSTR	MACS_TI	SAS_SA_Y_1 … SAS_SA_Y_3
Acquire the Y channel read-outs for SAS 1 to SAS 3		
MACS_INSTR	MACS_RD	THU_SA_DT_1 … THU_SA_DT_6
Receive the delay-time commands for THU 1 to THU 6		
MACS_INSTR	MACS_RD	THU_SA_OT_1 … THU_SA_OT_6
Receive the on-time commands for THU 1 to THU 6		

Subaddresses SAS_SA_X_1 to SAS_SA_X_3 are consecutive.

Subaddresses SAS_SA_Y_1 to SAS_SA_Y_3 are consecutive.

Subaddresses THU_SA_DT_1 to THU_SA_DT_6 are consecutive.

Subaddresses THU_SA_OT_1 to THU_SA_OT_6 are consecutive.

The format of the SAS sun position read-out is as follows:

Bit Fields	Value
Bits 0-7	Raw sun angle measurement in rad with quantization level of: SAS_Q
Bits 8-14	Unused – always zero
Bit 15	Sign bit: readout is positive if bit is equal to 1

The on-time and delay-time commands are expressed in milliseconds. Their format is as follows:

Bit Fields	Value
Bits 0-12	On-time/Delay-Time request in milliseconds
Bits 13-15	Unused – always zero

7.6 MACS Broadcasts

In addition to the instructions listed in the previous section, all MACS units can receive a MACS broadcast which is characterized by the MACS_BRDC extension.

7.7 MACS Traffic

MACS units are initialized when their AOCS unit proxy object is reset (this includes object construction) and when they are switched in as a result of a reconfiguration.

MACS units are assumed to be cold-redundant. When their reconfiguration manager takes them out of the active set, they are switched off. When it makes them active, they are switched on and initialized.

The MACS broadcast is sent at the beginning of each new AOCS cycle.

8 OPERATIONAL MODES

The following components exhibit mode-dependent behaviour in the AOCS prototype:

- failure detection manager
- failure recovery manager
- telemetry manager
- controller manager
- unit triggers
- mission mode manager

The mission mode manager has two operational modes – coarse sun pointing (CSP) and fine sun pointing (FSP) – corresponding to the two spacecraft mode of section 4.1.

With the exception of the telemetry manager, all the other mode-dependent components use mode managers of the follower type with the mission mode manager as master. They thus reproduce a situation of an AOCS software that has only two modes, the CSP and FSP modes.

The telemetry manager uses a cycling mode manager (see section 11).

8.1 Mode Change Actions

Mode change actions are not used. However, some of the mode managers performs actions associated to a change of mode as follows:

- The controller mode managers resets controller objects when they are switched in after a new mode is entered.
- The failure detection mode manager resets change objects when they are switched in after a new mode is entered.

8.2 Sensor and Actuator Usage

In CSP, attitude information is collected from the SAS and the gyro. In FSP, attitude information is collected from the FSS and the gyro.

In CSP, the thrusters are used as actuators. In FSP, the reaction wheels are used as actuators.

The control of the sensor and actuator power status is assumed to be done outside the AOCS. The AOCS is only responsible for sending a MACS bus initialize command to a unit when it is reset or when the AOCS is booted.

8.3 Control Laws

Proportional-Derivative (PD) controllers are used in both CSP and FSP.

On the X and Y axes the control is exercised on the attitude. On the Z axis, it is exercised on the angular rate.

The PD controllers on the X and Y axes are identical. Their parameters are: `mmm_XY_KP` and `mmm_XY_KP` where ‘`mmm`’ is the mode indicator (CSP or FSP). The parameters for the Z axis controller follow a similar naming convention.

Both the CSP and the FSP controllers operate with a control cycle frequency of 1 Hz.

9 DATA POOL ARCHITECTURE

The prototype AOCS software implements a single data pool: the *attitude data pool*.

The attitude data pool contains the following data:

Datum	Format
FSS read-out	Two Euler angles
GYR rate read-out	Scalar
SAS_1 read-out	Two Euler angles
SAS_2 read-out	Two Euler angles
SAS_3 read-out	Two Euler angles
Spacecraft torque	Three-vector
RW angular momenta	Four-vector
RW torque requests	Four-vector
THU on-time requests	Six-vector
THU delay-time requests	Six-vector
Atttitude estimate	Three Euler angles
Attitude set point	Three Euler angles
Attitude error	Three Euler angles

10 EVENT REPOSITORY ARCHITECTURE

The prototype AOCS software implements all the default event repositories foreseen by the AOCS prototype framework.

The repository sizes are as follows:

Event Repository	Size
Failure Event Repository	FAILURE_EVENT_REPOSITORY_SIZE
System Event Repository	SYSTEM_EVENT_REPOSITORY_SIZE
Mode Event Repository	MODE_EVENT_REPOSITORY_SIZE
Configuration Event Repository	FAILURE_EVT_REP
Reconfiguration Event Repository	RECONFIGURATION_EVENT_REPOSITORY_SIZE
Recovery Event Repository	RECOVERY_EVENT_REPOSITORY_SIZE
Telecommand Event Repository	TELECOMMAND_EVENT_REPOSITORY_SIZE
Manoeuvre Event Repository	MANOEUVRE_EVENT_REPOSITORY_SIZE
Change Event Repository	CHANGE_EVENT_REPOSITORY_SIZE

11 TELEMETRY

The AOCS prototype generates two telemetry frames corresponding to the two modes of the telemetry manager.

The two telemetry frames alternate in sequence.

The first telemetry frame contains images of the following AOCS objects:

- Failure event repository
- Configuration event repository
- Attitude data pool
- Mode event repository
- System event repository
- Recovery event repository
- Telecommand event repository
- Manoeuvre event repository

The second telemetry frame contains images of the following AOCS objects:

- Failure event repository
- Configuration event repository
- Attitude data pool

11.1 Telemetry Interface

The telemetry acquisition mechanism in the AOCS prototype is DMA based. The telemetry interface is as described in RD18.

12 FAILURE DETECTION

The failure detection tests are determined by the content of the consistency checkable and monitoring check lists associated to each operational mode of the failure detection manager.

The content of the consistency checkable list for the CSP mode together with the recovery actions associated to each consistency check is as follows:

Consistency Checkable Object	Recovery Action
Attitude data pool	Software reset

The content of the consistency checkable list for the FSP mode together with the recovery actions associated to each consistency check is as follows:

Consistency Checkable Object	Recovery Action
Attitude data pool	Software reset

The content of the monitoring check list for the CSP mode together with the recovery actions associated to each monitoring check is as follows:

Monitoring Check	Recovery Action
Delta change check on FSS X channel read-out	FSS reconfiguration
Delta change check on FSS Y channel read-out	FSS reconfiguration
Out-Of-Range check on FSS X channel read-out	FSS reconfiguration
Out-Of-Range check on FSS X channel read-out	FSS reconfiguration
Delta change check on GYR rate channel read-out	GYR reconfiguration
Out-Of-Range check on GYR rate channel read-out	GYR reconfiguration

The content of the monitoring check list for the FSP mode together with the recovery actions associated to each monitoring check is as follows:

Monitoring Check	Recovery Action
Delta change check on FSS X channel read-out	FSS reconfiguration
Delta change check on FSS Y channel read-out	FSS reconfiguration

Out-Of-Range check on FSS X channel read-out	FSS reconfiguration
Out-Of-Range check on FSS X channel read-out	FSS reconfiguration
Delta change check on GYR rate channel read-out	GYR reconfiguration
Out-Of-Range check on GYR rate channel read-out	GYR reconfiguration
Out-Of-Range check on X-axis attitude error	fallback to CSP
Out-Of-Range check on Y-axis attitude error	fallback to CSP

13 FAILURE RECOVERY

The AOCS prototype implements the same three failure recovery strategies in both CSP and FSP:

- Command software reset if a configuration error has been detected in the last cycle
- Command software reset if the number of failure events created in the last cycle in the failure recovery repository exceeds MAX_FAIL_EVT
- Perform local recovery actions associated to each failure event created in the failure event repository in the last cycle

The recovery actions associated to failure events are always the null recovery action except for those associated to the monitoring and consistency checks listed in section 12.

14 TELECOMMANDS

The telecommands loaded with the AOCS software and therefore available for execution by data-only telecommands are:

- Telecommand to change the operational mode of a mode manager component
- Telecommand to change the telemetry format of a telemeterable component
- Telecommand to reconfigure a reconfiguration manager
- Telecommand to load and configure an attitude slew manoeuvre in the manoeuvre manager
- Telecommand to load a parameterless manoeuvre in the manoeuvre manager

The first three telecommands are provided both as simple telecommands and as transaction telecommands. The last two are provided only as simple telecommand.

Dynamical loading of telecommand code is not foreseen for the AOCS prototype.

14.1 Telecommand Interface

The telecommand acquisition mechanism in the AOCS prototype is DMA based. The telecommand interface is as described in RD18.

15 MANOEUVRES

Only one type of manoeuvres is available in the AOCS prototype. This is a manoeuvre to perform a single-axis attitude slew. The manoeuvre implements a linear slew updating the attitude set point at time t according to the following algorithm:

```
att_set_point = att_start + (t-t0)*(att_end-att_start)/(tf-t0)
```

where `att_end`, `tf` and `t0` are manoeuvre parameters defining the target attitude for the manoeuvre, the end time and the start time for the manoeuvre, respectively. `att_start` is the attitude at the time when the manoeuvre starts.

The manoeuvre is aborted if the attitude error size exceeds a certain threshold which is also a settable manoeuvre parameter.

Three attitude slew manoeuvre objects are instantiated on board so that a maximum of three attitude slews can be running at any given time.

16 SCHEDULING

The AOCS prototype is based on a simple non-preemptive cyclical scheduler. It consists of the following tasks:

- the unit acquisition trigger unit
- the controller manager
- the failure detection manager
- the failure recovery manager
- the manoeuvre manager
- the telemetry manager
- the DMA telecommand loader
- the telecommand manager
- the unit commanding trigger unit

All tasks are represented by components implementing the `Runnable` interface (active objects).

The trigger unit at the beginning of the cycle sends the MACS instructions to acquire data from the external units (all the sensors and the reaction wheel speed). The trigger unit at the end of the cycle sends the MACS instructions to send commands to the external units (the actuators).