S/0-18

/6-/2/2

AN APPROACH TO THE DESIGN AND IMPLEMENTATION OF SPACECRAFT ATTITUDE CONTROL SYSTEMS

James R. O'Donnell, Jr., Ph.D.
David J. Mangus
Goddard Space Flight Center, Code 572
Greenbelt, Maryland USA 20771

Over 39 years and a long list of missions, the guidance, navigation, and control (GN&C) groups at the Goddard Space Flight Center have gradually developed approaches to the design and implementation of successful spacecraft attitude control systems. With the recent creation of the Guidance, Navigation, and Control Center at Goddard, there is a desire to document some of these design practices to help to ensure their consistent application in the future.

In this paper, we will discuss the beginnings of this effort, drawing primarily on the experience of one of the past attitude control system (ACS) groups at Goddard (what was formerly known as Code 712, the Guidance, Navigation, and Control Branch). We will discuss the analysis and design methods and criteria used, including guidelines for linear and nonlinear analysis, as well as the use of low- and high-fidelity simulation for system design and verification of performance. Descriptions of typical ACS sensor and actuator hardware will be shown, and typical sensor/actuator suites for a variety of mission types detailed. A description of the software and hardware test effort will be given, along with an attempt to make some qualitative estimates on how much effort is involved. The spacecraft and GN&C subsystem review cycles will be discussed, giving an outline of what design reviews are typically held and what information should be presented at each stage. Finally, we will point out some of the lessons learned at Goddard.

INTRODUCTION

Throughout its history, the Goddard Space Flight Center has had a number of attitude control system (ACS) branches and organizations devoted to the design, development, testing, and operation of the attitude control and determination subsystems of spacecraft. During the many years and many projects with which these groups have been involved, a number of practices, approaches, and lessons learned have been developed that have led to a great many successful missions.

With the recent reorganization of engineering groups at Goddard, a number of attitude control and navigation groups have been merged and combined within the new Guidance, Navigation, and Control Center (GNCC). The approaches and material discussed in this paper primarily reflect the heritage of only one of the antecedent groups to the GNCC, what was formerly Code 712, the Guidance, Navigation, and Control Branch. Code 712 had primary responsibility for the mid-range and larger spacecraft designed and

built at Goddard. Other groups, also part of the new GNCC, designed the ACS subsystems for the smaller missions or were involved in ACS on-orbit operations. A paper written from the points of view of these groups would reflect many of the same general approaches, though it would likely differ in emphasis.

The goal of this paper is to begin the work of documenting the different aspects of the ACS subsystem design process. The paper will cover many topics very broadly in an attempt to give an overview of our work and some feeling for our design approach. It is hoped that this effort will continue in a much more complete and rigorous way, so that the ACS design heritage and expertise of the Goddard ACS groups can be preserved for the future.

ACS SUPPORT FOR SPACECRAFT PROJECT PHASES

A typical spacecraft project goes through at least five general phases, four of which require the direct, full participation of ACS engineers. These first four phases are the spacecraft conceptual design, development, integration and test, and launch and early operations. Once a spacecraft is on-orbit and operating, ACS involvement usually is limited to support for special events such as orbit maneuvers, anomaly resolution, and participation in end-of-life and other engineering tests.

Following is a list that summarizes many of the tasks that are performed by members of the ACS subsystem throughout these phases of a spacecraft's life.

1. **Spacecraft ACS Conceptual Design**
   - Support GN&C systems engineering in defining high-level analysis support
   - Support GN&C systems engineering in determining contractor support
   - Review project-level requirements with scientists
   - Develop GN&C mission-level requirements from project-level requirements
   - Define ACS hardware requirements
   - Define ACS software requirements
   - Define attitude knowledge and control requirements
   - Define trajectory requirements
   - Design GN&C mission scenario
   - Develop control mode block diagrams
   - Develop control mode error budgets
2. **System Development**
   - Verify control mode rigid body stability margins
   - Develop a low-fidelity time domain simulation (LoFi)
   - Verify control mode pointing performance using LoFi
   - Perform structure modal reduction analysis
   - Perform control mode flexible body stability margins analysis
   - Develop high-fidelity models of selected sensor and actuator hardware
   - Develop a nonlinear high-fidelity time domain simulation (HiFi)
   - Verify control mode error budget using HiFi
   - Verify control mode transitions using HiFi
   - Present at the Critical Design Review
   - Develop the ACS Algorithm Document
   - Support the definition of the flight software test facility
   - Write software test procedures that verify control mode requirements
   - Perform and evaluate results of control mode software testing
   - Support hardware procurement and hardware and software design reviews
3. **Integration and Test (I&T)**
   - Develop and support spacecraft-level ACS hardware aliveness tests
   - Develop and support spacecraft-level ACS hardware functionality test procedures
   - Develop and support spacecraft-level ACS hardware phasing test procedures

- Develop and support spacecraft-level ACS end-to-end tests
- Develop on-orbit ACS procedures
- Develop and support ACS level Comprehensive Performance Tests

4. **Launch and Early Operations**
   - Determine on-orbit sensor and actuator calibration requirements
   - Develop mission operation center ACS telemetry page layouts
   - Support pre-launch mission operations
   - Support launch and early on-orbit activities
   - Support on-orbit sensor and actuator calibrations

The remainder of this paper will discuss these phases, and the different work that occurs in each, in some more detail. The main concentration, again reflecting the experience of the authors and Code 712, will be on the development and I&T phases, with a little discussion on the systems engineering and conceptual design phase.

SYSTEMS ENGINEERING AND DESIGN

ACS support for spacecraft development begins in the initial systems engineering and design effort, which very often begins a number of *years* before a project is even approved. There are usually a variety of study and "pre-Phase A" efforts involved in the very early spacecraft design; once a project is approved, it enters a much more formal period of development and test.

During the early systems efforts, systems engineers and those who work directly with the ACS subsystem begin the initial design task, taking the science and other mission requirements and using them to design ACS requirements and a subsystem concept that will allow the spacecraft to successfully achieve its science objectives. This early ACS design comprises the following steps:

1. **Mission Concept Design:** In this design phase, the object is to develop a concept that will allow for the successful completion of the mission's science objectives, within the imposed budgetary, time, and other constraints. This mission concept encompasses all phases of the project, from initial design and development, through test, launch, and operations. Specific to the ACS, an initial design of the subsystem is created—including control modes and preliminary hardware sensor and actuator complement—that will allow the mission concept to be successfully implemented.

2. **ACS Level Requirements:** Once the mission concept is developed, the mission-level requirements must be used to generate subsidiary requirements for each subsystem. For the ACS, this will typically mean requirements for pointing accuracy and stability, momentum management, and orbit maneuvering and maintenance; as these requirements are developed and refined, they can have a direct impact on the subsystem design, both algorithmic and hardware.

3. **Error Budgets:** With numerical ACS subsystem requirements in hand, it is necessary to develop error budgets that parcel out the required performance goals and allowable errors to the different parts of the subsystem. These budgets will define the type and quality of ACS hardware required, as well as imposing performance and stability requirements on the control algorithms developed in the design process.

4. **Trade Studies:** It is usually necessary to iterate on all of the above steps, generating a number of options and possibilities for the different aspects of the design. Decisions made at the subsystem level, both in the ACS subsystem and others, can have potential impacts in other subsystems and on the system as a whole. The object of the early stages of the design is for the systems engineers, aided by specialists in the different subsystem areas, to generate a viable preliminary design and attainable set of requirements so that the design and development effort can continue with an expectation of success.

ACS ANALYSIS AND DESIGN

In the ACS analysis and design phase, the requirements and mission concept generated in the early systems engineering effort is turned into reality—at least a mathematical and algorithmic reality, at this point. The first step is to define the control modes—and relationships between modes—that can implement the mission concept defined in the conceptual design of the spacecraft ACS. Then, for each mode, a linear controller design and analysis must be performed.

For the majority of spacecraft attitude control systems, initial design is performed assuming a PD or PID controller. Because of the relatively benign disturbance environment for most space missions, this simple controller usually proves to be sufficient. Using a PD or PID controller and a linear J/s^2 plant to model the spacecraft, beginning the design of controller gains and analysis of ACS performance is very straightforward. As the physical design of the entire spacecraft becomes more mature, and mathematical models of it more complete, a flexible mode analysis is done to ensure that the spacecraft ACS will not excite any uncontrolled oscillations during operation. Throughout the design process, simulations are developed and run to verify the time-domain performance of the spacecraft.

A final aspect of the ACS algorithmic design does not apply specifically to the spacecraft performance within a specific mode, but looks at the management of angular momentum across all modes. Depending on the mission, all spacecraft will encounter a variety of disturbance inputs—aerodynamic, gravity gradient, solar pressure, and magnetic, to name the most common—that, over time, can cause a buildup of system angular momentum within the spacecraft. A system for managing and off-loading this momentum must be designed, as it will eventually affect the performance of the spacecraft ACS.

Control Mode Definition

Figure 1 shows an example control mode diagram from the MAP spacecraft. The six modes, five in the spacecraft main ACS processor and one in the separate attitude control electronics (ACE) box, were designed to meet the requirements derived from the mission requirements and concept for the MAP ACS subsystem.

Linear System Design and Stability

It is during the linear ACS design and analysis phase of a project that the work most commonly associated with the ACS *analyst* is done. For the large majority of spacecraft, the control laws used as the basis for ACS design remain the same as those used 20 years ago. The initial design and stability analysis is mainly concerned with using the available tools to decide what control law is needed, determine the controller gains and other parameters needed to implement that control law, and then to verify its performance and stability. As more information—such as system inertia matrices and flexible mode analyses—becomes available, further analysis is performed to verify that sufficient performance and stability margins still exist.

The bulk of the engineering judgement and expertise of the ACS analyst with respect to the linear design come into play in two main areas. First, the initial design must include sufficient performance and stability margins so that, as more information about the system becomes known and more fidelity is included in the design models, the design continues to satisfy all performance and stability criteria. Second, the control law and linear ACS design must be created and translated in such as a way so that it can be turned into physical hardware and software on the spacecraft.
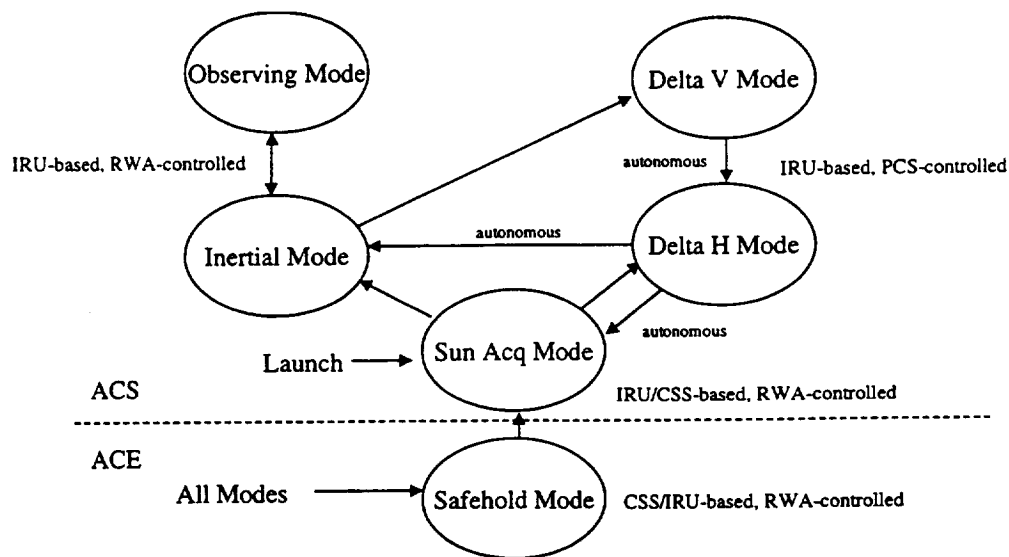
Figure 1 MAP Control Modes

Design and Analysis Tools. Many of the design and analysis concepts used by today's ACS analyst haven't changed from those used his or her predecessor, 20 or 30 years ago. The same collection of methods used to characterize the performance and stability of systems modeled with linear equations has been used in the aerospace industry for as long as it has been around. Of course, the way these tools are implemented has changed quite a bit with the advent and increasing power of the computer. Today's analyst may be creating root locus, Bode, and Nichols plots much like analysts of the past, but he or she is generating them a lot more quickly.

Figure 2 shows a collection of plots that represent some of the typical means of designing and analyzing attitude control systems modeled with linear equations. Root Locus plots allow the designer to work directly with the open- and closed-loop poles and zeros of the linear system; well-known relationships can then be used to derive frequency- and time-domain characteristics, such as natural frequency and damping ratio, rise time and settling time, for the resulting system. Bode and Nichols plots (as well as Nyquist plots, an example of which is not shown) are alternate methods of displaying the same frequency-domain information and characteristics of a system. The most important of these characteristics, as relates to system stability, are the system gain and phase margins. These margins give a measure of how much more or less gain and how much more phase lag a system can handle before it goes unstable. Adequate margin is needed when designing a system modeled with linear equations to ensure that the real system will remain stable and have acceptable performance in actual operation.

The final design tool used by the ACS analyst is the time-domain simulation. Through a variety of simulations of varying degrees of fidelity, the performance of the control system is analyzed and verified.

As mentioned above, many of the design techniques used by ACS designers has remained the same. The way in which the designer generates and makes use of these tools has changed, and is beginning to change a lot more quickly, as computers become faster. For instance, using a software tool such as the Interactive Control Design Module from Integrated Systems, the linear equations for a system and a baseline controller can be set up, and the controller parameters can be changed and the resulting system performance viewed in real-time.

107
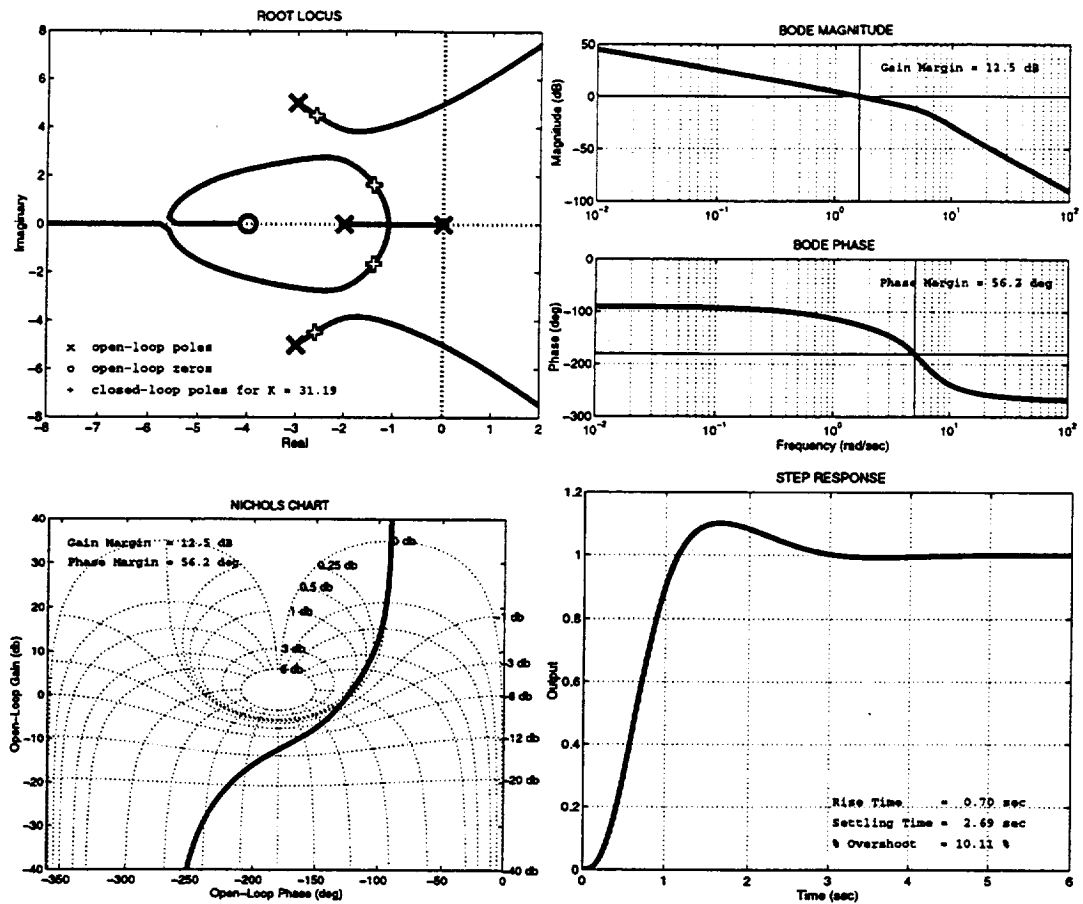
Figure 2 Linear Analysis and Design Techniques

Rigid Body Stability. The initial ACS design and analysis is usually done with a simple linear model of the spacecraft, using only rigid body dynamics. At this stage of a project, it is usually not possible to accurately model any better than this. It is not until the physical design of spacecraft bus and instrument become more mature that details of the flexible body characteristics of the system become available.

There are a number of design criteria that can be applied to a rigid body design. Of course, most of the time-domain criteria are dictated by the requirements of the mission; these include things like slew rate and pointing performance. Controller designs must first satisfy these requirements. However, there are a number of criteria that determine the stability of a system that also must be satisfied. The criteria used are the gain and phase margins of the system[1]. The margins that we typically use, which must be found with respect to any commandable gain within the control system (i.e., controller gains, reaction wheel or other actuator scale factors, etc.), are a gain margin of 12 dB and a phase margin of 40°.

Flexible Body Stability. Flexible body analysis is generally performed starting with a NASTRAN (or other such) model of a system that gives the flexible mode frequencies and modal gains of a system. This list of modes is reduced, based on the modal frequencies and gains compared to the bandwidth of the control system, into a number of modes used for linear analysis. These modes are then included in the linear model of the system as a series of second-order systems of the form:

$$\frac{K_i}{s^2 + 2\varsigma\omega_i s + \omega_i^2} \tag{1}$$

108

The parameters K_i and ω_i in Eq. (1) are the modal gain and frequency of mode i, respectively. The parameter ς is the damping ratio of the flexible modes, which is conservatively set to 0.001 (this translates to an impulse response that takes approximately $5\tau = 5/\varsigma\omega_i = 5000/\omega_i$ seconds to die out^2). With the flexible modes in place, stability analysis is performed using the same desired gain and phase margins as used above.

Performance

Along with determining the stability of a system, generally using frequency-domain techniques such as Bode or Nichols plots, the performance of a system must be determined using time-domain simulations of the system. Generally, three types of simulations are created and used during a project for analysis and design. The first of these is a simple linear simulation, using the same linear model as that used for frequency-domain analysis. Time-domain performance of such a model can be calculated analytically by the same design package that is used to create Bode or Nichols plots, and is typically used as a first cut during the early stages of the design.

Aside from simple linear models of a system, nonlinear models and simulation tools are used with both low-fidelity (LoFi) and high-fidelity (HiFi) models of a system

Low Fidelity Simulations. Low-Fidelity (LoFi) simulations, like simple linear models, are typically used during the early stages of an ACS design and analysis effort to broadly characterize the performance of the system. LoFi simulations differ from linear models in that they typically include some nonlinear elements, particularly mixed discrete and continuous systems to model the physical environment and dynamics of a spacecraft along with the discrete controller.

Typically, the LoFi simulation is used during the early design stages when there is likely to be a lot of iteration back and forth between design and performance verification, and so a quick turnaround is highly desirable. The intent of the LoFi is to show broad performance characteristics, enough to verify that the design can meet the pointing, slew rate, and other criteria.

High Fidelity Simulations. Whereas a LoFi simulation is primarily a design tool, a high-fidelity (HiFi) simulation is mainly used for design verification. HiFi simulations are created by including as much detail into the simulation as possible, including such things as sensor and actuator performance and noise models, environmental disturbances, quantization error, and even the ability to model non-ACS specific events such as ACS processor warm and cold restarts. Also, as will be discussed in the ACS Flight Software section later in this paper, automatic code generation tools can even allow the HiFi to be used to generate actual flight software.

Figure 3 shows an example of a LoFi and HiFi spacecraft simulation. Notice that the LoFi shows a fairly accurate portrayal of the "macro" performance of the control system. The HiFi, which is plotted on a much tighter scale, gives a better view on the "micro" level, showing the details of the system response in the presence of noise and other disturbances.

Momentum Management

Design and analysis of the momentum management component of the ACS begins in the initial study phase of a spacecraft design, and continues as the spacecraft design matures and the mathematical models used for it continue to be developed. The following considerations and rules of thumb are used in its design:

1. Identify the external torques that contribute to a build-up of system momentum. Typically, these torques are gravity gradient, aerodynamic, and solar pressure. In low-Earth orbits, gravity gradient and aerodynamic torques are the most significant; at higher orbits, the most significant external torque is usually from solar pressure.
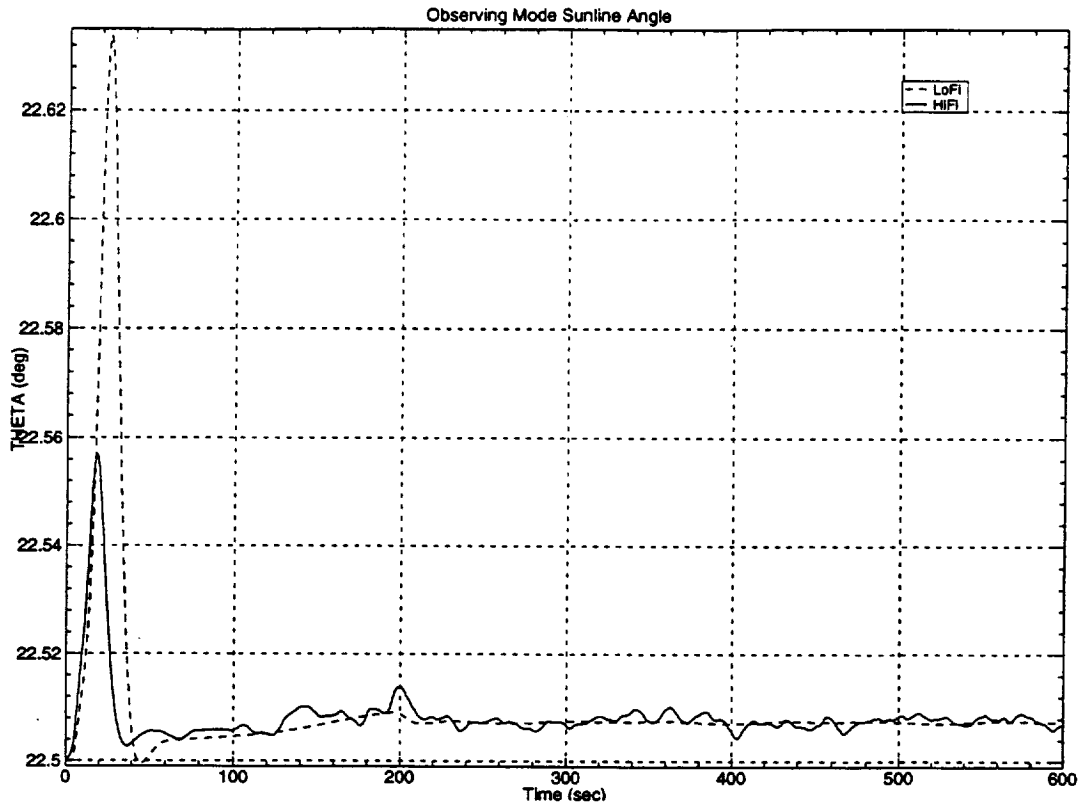
Figure 3 Low- and High-Fidelity Simulations

2. Select a momentum unloading control law. The two control laws generally used are referred to as "B dot" and "HxB". **B dot** does not require any hardware other than magnetic torquer bars and a magnetometer, but leaves a residual spacecraft body rate equivalent to one or two revolutions per Earth orbit, and will not dump momentum stored in the wheels. HxB, on the other hand does not leave the spacecraft with a residual rate, but it also requires system momentum information (typically from gyros and reaction wheel tachometers).

3. Using past missions in a similar orbit, analysis of worst-case environmental disturbance torques, and LoFi simulations as a guide, make a first cut at magnetic torquer bar sizing.

4. Select torquer bar sizes including 100% margin. Inertia ratios, and other parameters that influence the spacecraft response to environmental torques, will typically rise by more than 30% from the initial sizing studies to launch.

5. Test the final design using the HiFi simulation. It is best to use a 36-hour test, so that a full 24 hours is covered to allow the Earth magnetic dipole to rotate in inertial space, with an additional six hours of overlap on each side.

A similar approach, using similar margins, is taken when magnetic momentum unloading impractical or impossible. Two other possible approaches to momentum unloading are using thrusters (which is very quick and efficient, but requires expendable fuel) or, in higher orbits where the dominant disturbance torque is solar pressure, by trimming the orientation of the solar panels with respect to the sunline.

ACS HARDWARE

As the ACS is being designed by analysts, other ACS subsystem engineers are selecting and procuring the hardware to be used to implement that design on the spacecraft. The hardware needed for the ACS

110

subsystem can be divided up into three categories: flight hardware shared by the ACS and other spacecraft subsystems, the ACS sensor, actuator, and other electronic hardware, and the hardware needed on the ground to integrate and test the ACS system.

Depending on the design of a given system, the ACS will usually be implemented within the main spacecraft processor, within a separate attitude control electronics (ACE) box, or sometimes both. The main processor, along with the hardware and software that comprise the command and data handling (C&DH) subsystem, are integral parts necessary for the successful operation of the ACS. Similarly, the attitude control electronics, used as an interface between the control algorithms implemented in either the main processor or a dedicated ACE processor and the sensor and actuator hardware. Also, instrument data is sometimes used for fine position sensing within the ACS.

Flight Sensor Hardware

There is a wide variety of ACS sensor and actuator hardware available for spacecraft missions—each piece of hardware has different characteristics of performance, cost, lifetime, and other criteria, that make it applicable to a subset of the possible missions. In this section of the paper, we will give examples of some of the major types of sensors and actuators. At the end of the section we will show a table containing some representative mass, power, and performance numbers for some of the sensor types discussed.

Earth Sensor: Earth sensors detect the Earth's horizons (actually, an infrared radiation band from the CO_2 layer slightly above each horizon) as seen from space to provide two axes of attitude information with respect to the geodetic nadir vector from the spacecraft to the Earth. Earth sensors cannot measure the yaw angle about this vector. These sensors are very often used as the primary attitude sensors for Earth-pointing spacecraft; even when the pointing requirements of the mission exceed the capabilities of an Earth sensor, they are often used for Earth acquisition after launch and other maneuvers.

There are two general classes of Earth sensors. The first type, scanning Earth sensors, use a moving optical head to detect where the horizon is. Static Earth sensors, on the other hand, are built to operate at a given altitude range and have a fixed field-of-view designed to intersect the Earth horizon at one or more points. In general, scanning sensors are heavier, use more power, and cost more (in addition to causing attitude disturbances that may affect the science payload), but also have a larger range of operation and greater pointing performance.

Digital Sun Sensor: Digital sun sensors are generally used to detect the orientation of a spacecraft with respect to the vector from the spacecraft to the sun. Like an Earth sensor, sun sensors cannot measure the orientation of the spacecraft about this vector. Digital sun sensors can be used to give two axes of information, and are generally used with other attitude measurements as part of a general attitude determination algorithm, or to provide attitude updates to a Kalman filter.

Inertial Reference Unit: Inertial reference units are gyro/accelerometer packages that can be used as an angular rate and acceleration sensor (to save cost, weight, and power, the accelerometers are often left out, unless there is a structural resonance issue). They can also be used to propagate an attitude estimate between measurements from an attitude sensor. Gyros are available in a wide range units, varying considerably in mass, power, and capabilities.

Star Tracker: Star trackers are use to detect and track stars. By correlating measurements of the line of sight vectors to multiple stars, the spacecraft's current orientation with respect to an inertial reference frame can be determined (the particular inertial frame used depends on that used to specify the star positions).

Most star trackers currently operate by providing information to the spacecraft processor about line-of-sight vectors and magnitudes to detected stars. Algorithms in the processor, using an onboard star catalog, identify the stars detected and use their defined positions to calculate the spacecraft attitude. In the event of a spacecraft where this information is only needed for attitude knowledge, and not attitude control, this processing may also be done on the ground. However, some of the newest star trackers now becoming available are so-called "quaternion output" trackers—they have their own processor and built-in star

catalog, and can directly output a quaternion expressing the orientation of the star tracker boresight with respect to an inertial reference frame.

	Mass (kg)	Power (W)	Performance
Earth Sensor (static)	0.69	0.5–2	20° field of view 0.25° accuracy
Earth Sensor (scanning)	2.5	8	45° field of view 0.05–0.1° accuracy
Digital Sun Sensor	2	1.05	±32° field of view 0.017° accuracy 0.0039° resolution
Inertial Reference Unit (two-axis unit)	1.6–5.8	7.5–12	0.1–1 arcsec/pulse
Star Tracker	8.1	12	8×8° field of view 3 arcsec accuracy

Other Sensors: In addition to the sensors discussed above, which are the typical sensors usually used by the ACS mission mode controller, there are a number of other sensors that are generally included or can be used on a spacecraft. These sensors fall into two general categories:

First, there are sensors used for a special purpose on a spacecraft, other than a normal mission mode control. Two examples of this would be the course sun sensor, usually used as a part of a minimum hardware sun acquisition control mode, and the three-axis magnetometer, usually used along with magnetic torquer bars for momentum unloading. (It is interesting to note that algorithms, such as the "Contingency Mode" developed for the TRMM spacecraft, have been developed and implemented that use magnetometer measurements, along with measurements from other sensors, such as digital sun sensors, to generate a mission mode attitude estimate.)

Second, there are a number of new technology attitude sensors that are in development and/or at the experimental stage. These include the use of GPS signals in a differential mode to generate attitude information, and a number of integrated star tracker/inertial reference unit sensors that generate both attitude and attitude rate information.

Flight Actuator Hardware

Most spacecraft use one of two types of actuators for most of their flight actuator requirements. Propulsion systems and thrusters are included on a spacecraft to perform orbit maneuvering and stationkeeping, as well as for momentum management on spacecraft where magnetic torquer rods are not sufficient (or usable, for non-Earth orbiting or high-Earth orbiting spacecraft). The ACS groups work closely with the propulsion groups at Goddard to design a thruster system, where needed, that will satisfy all spacecraft requirements for attitude, orbit, and momentum management. In general, though, most spacecraft require reaction or momentum wheels as their primary actuator.

Reaction Wheel: A reaction wheel is used to apply or remove a rate from a spacecraft by making use of the conservation of angular momentum. When torque is applied to a reaction wheel to get it to change its rotation rate about a given axis, a corresponding torque and angular rate change is generated in the spacecraft in the opposite direction. Two of the most important ways that reaction wheels are characterized is through their momentum capacity and their torque authority—in general, larger wheels have more capacity and can generate a larger torque, at the expense of more mass, power, and cost.

	Mass (kg)	Power (W)	Performance
Reaction Wheel "Type A"	2.55	5.5–9 (orbit average) 25 (peak)	0.012–0.02 Nm torque authority 4 Nms momentum capacity
Reaction Wheel "Type E"	10.5	15–40 (orbit average) 280 (peak)	0.3 Nm torque authority

Momentum wheels are similar to reaction wheels; generally, they provide lower torque, but have a higher efficiency. Lastly, in addition to thrusters and wheels, magnetic torquer bars are often used by spacecraft in low-Earth orbit for momentum management.

Test Hardware

In addition to the flight hardware that must be acquired to implement an ACS design on a spacecraft, there is a large amount of additional hardware that is required for ground test purposes. Figure 4 shows an example of the hardware used for a hybrid dynamic simulator (HDS), used to provide ground-test capability for a spacecraft, both software algorithms and flight hardware, from low-level testing through final integration and test.
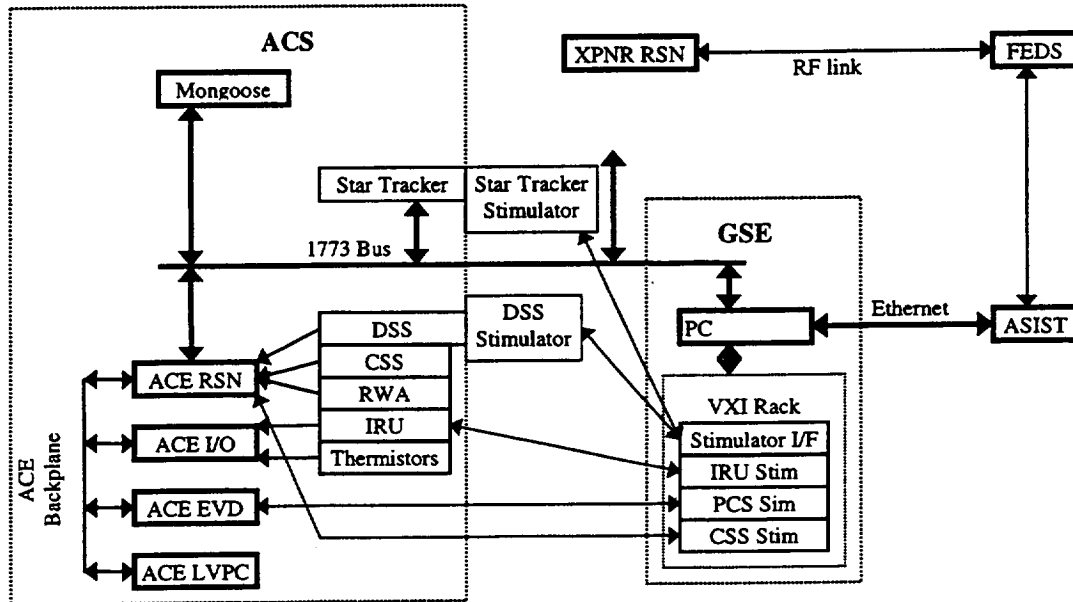

Figure 4 HDS Hardware for ACS Testing

Multiple hardware test setups are typically required for many missions, allowing the ability to concurrently test the main spacecraft processor, the independent attitude control electronics (if present), as well as the ability to stimulate all flight hardware and hardware interfaces. The production of breadboard and engineering test units (ETU) for many of the flight components adds to the cost and effort involved. Depending on how the test effort is scheduled, very often completely different test "strings" are required to allow for testing of ACS and C&DH functionality separately. When all of this additional hardware required for the testing effort is considered, it becomes a significant fraction of the cost and effort associated with the flight hardware.

ACS FLIGHT SOFTWARE

There are three major phases in the production of ACS flight software used to implement the control laws developed to support the spacecraft mission. The first phase bridges the gap from the ACS design described above to the flight software developers, usually through the means of an algorithm document. Next, the ACS algorithms, along with all of the other necessary flight software functions, are developed for the specific processors and hardware architecture selected for the spacecraft. Finally, perhaps most importantly, software (and hardware, as discussed above) is developed and run to fully test the functions of the flight software.

ACS Algorithm Document

For the TRMM project, ACS analysts converted the algorithms needed to implement the controller into FORTRAN-like syntax used in the TRMM ACS Algorithm Document[3]. This document was then provided to the ACS flight software developers who used it as a guide to develop the C language routines necessary to implement the algorithms on the spacecraft main and ACE processors.

One of the lessons learned from the TRMM experience is that the FORTRAN vs C syntax used for the algorithm document and flight software, respectively, hampered the ability of the developers to correctly interpret the algorithm document, as well as making it more difficult for the ACS designers to verify the correctness of the code. Using similar syntax—which in this case would have meant using C-like syntax in the algorithm document—would have made the process a lot easier.

Taking advantage of some of the advances in the technology of the control system design tools, a slightly different approach to that used by TRMM is now possible. Products such as DocumentIt from ISI allow documentation to be generated automatically from a system model, such as a HiFi simulation. The documentation generated for the MAP ACS uses the software tool's ability to create output meant for display on the Web.

The ability to automatically generate documentation is only a small part of what is becoming possible with the newest generation of ACS design tools. As will be discussed in the next section, there are several tools available now that will automatically generate flight software from a system model, thus skipping the algorithm to algorithm document to flight software translation process altogether.

Automatically-Generated Code

Figure 5 shows an example block diagram from the MAP HiFi simulation, depicting the Observing Mode controller. Instead of translating this controller into a written algorithm, which would then be coded by the ACS flight software developers, by using ISI's AutoCode ACS development tool, flight code for this controller can be automatically generated.

There are a number of potential benefits to automatic code generation. Primarily, once confidence in the code-generation tool is established, it should strengthen the testing effort. Because it is possible to generate some of the flight code much more quickly using this method, it is possible to begin testing earlier, and thus test more thoroughly. Also, because the algorithms are proven within the simulation environment first, before the code is generated, a great deal of lower-level testing can be avoided.

The MAP program is one of the first at Goddard to use automatically generated flight software. Because of this, the scope of what parts of the ACS were chosen to be AutoCoded was limited; even so, approximately 1/3 of the MAP ACS flight software was automatically generated. The consulting group at ISI, which has a lot more experience using automatic code generation, and the ISI tools in particular, has reported even greater percentages of automatically generated code and improvements of the flight software design cycle for the MSTI-1, MSTI-2, and MSTI-3 programs[4].

Development vs Testing

Based on the experiences that we have had across many missions, it is safe to say that the flight software testing effort is always underestimated. Just as software is developed to implement ACS algorithms for flight, software test procedures are needed to test that software throughout the development and integration and test phases of the project. The relative amount of effort needed for ACS flight software development vs testing is roughly a 50/50 split between the two. This includes the manpower and time to develop the software, either flight or test, and to run the tests; it doesn't include ACS analyst support, but that also tends to be pretty evenly split between the two activities.

The relative effort assessments discussed above do not include that involved for configuration management and maintenance of a system for tracking ACS parameters and for filing discrepancy reports (DRs) for the different ACS subsystem components. One of the lessons learned from past programs that is currently being applied is the use of a central database for maintaining the DRs and parameters used throughout the ACS subsystem: HiFi simulation, flight software, and HDS. Because of this, it is fair to consider this effort as a general overhead important to both the development and testing effort.

DESIGN REVIEW CYCLE

Each spacecraft goes through a number of design reviews, each of which has a different emphasis and reflects a different stage in the design process. While the specific names of the reviews for each project may vary, the general topics covered at each stage tend to be the same. Some of the typical reviews and topics covered, as they relate to the ACS subsystem, over the course of a project are discussed in this section.

There is some overlap from one review to another, and a number of topics that will be covered in each. In particular, every review will discuss design changes (and their impacts) and action items (and their resolutions) since the last review, as well as outstanding issues and concerns.
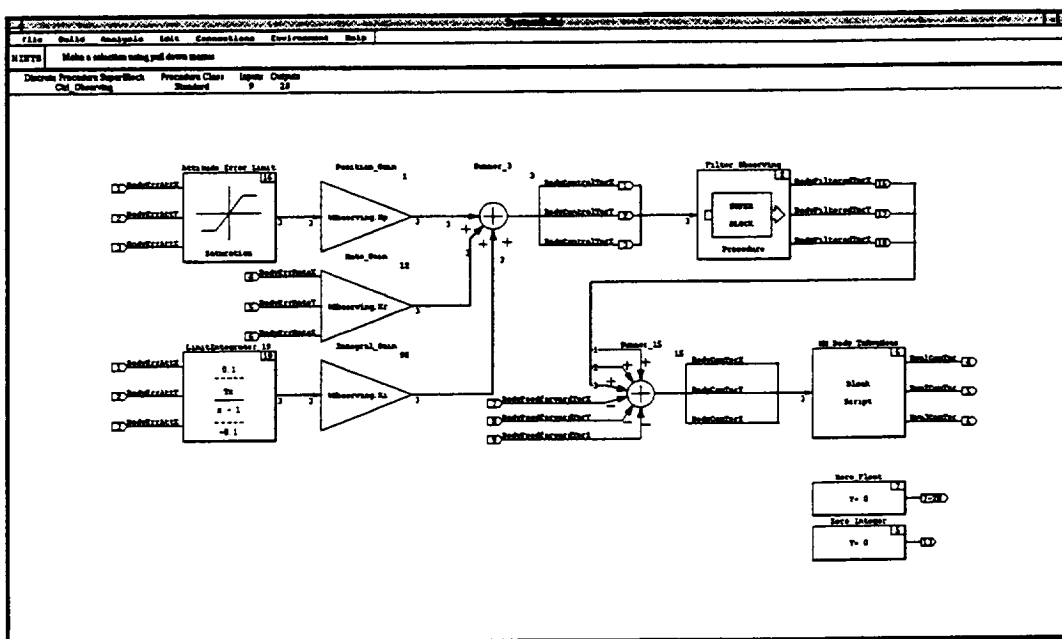

Figure 5 MAP Observing Mode Controller to be AutoCoded

Preliminary Design Review (PDR)

The first formal review is usually the Preliminary Design Review, or PDR. This is typically conducted after enough time has passed for a first cut at an ACS subsystem design has been done. The purpose of the initial design and review is to develop a preliminary design for the complete ACS subsystem and to determine the viability of the mission concept. If the mission has any "show stoppers"—mission requirements that cannot be met within the budgetary, time, or other constraints imposed on the project—they should be identified by the PDR. Also, major design issues should be identified. A list of topics typically covered in a PDR is as follows:

1. Requirements
2. Subsystem Analysis

- Analyses Performed
- ACS Modes
  - Descriptions
  - Stability
  - Performance
- Other Topics (Attitude Determination/Error Budgets, Momentum Management)
3. Hardware
   - Procured Components
   - In-House Development
4. Software
5. Operations
6. Testing

Critical Design Review (CDR)

The Critical Design Review (CDR) is the last subsystem-specific review conducted. By the time it is conducted, the ACS design should be completed, analyzed, and tested to the point that it can be said and demonstrated with confidence that the design will meet all requirements. Issues identified at the PDR should be addressed and closed at the CDR. The design of other parts of the spacecraft should have matured by subsystem CDR so that flexible mode analysis is possible. The CDR will cover much of the same material as the PDR, though the design should be more mature and finalized.

1. Requirements
2. Subsystem Analysis
   - Status
     - Stability Analyses
     - Flexible Mode Analyses
     - ACS Mode Description and Performance Summaries
   - Other Topics (Attitude Determination/Error Budgets, Momentum Management)
3. Noncompliance Summary/Open Issues
   - Hardware
   - Software
4. Failure Detection and Correction
5. Operations
6. Testing

Once all subsystems have had their CDRs, there is a spacecraft-level CDR that covers the complete spacecraft, at a higher level. It is through this review, either on its own or with a separate confirmation review, that it is decided whether or not the spacecraft design is complete and good enough to justify continuing with the integration and test process.

Pre-Environmental Review (PER)

A Pre-Environmental Review (PER) is conducted immediately prior to the full spacecraft going into environmental testing. The purpose of the review is to ensure that all outstanding issues from previous reviews and I&T have been closed, and that a plan is in place for successfully testing spacecraft aliveness and functionality as it goes through the different environments. Items typically covered in a PER include:

1. Hardware Qualification and Testing
   - For Each Component:
     - Hardware Description and Part Number
     - Component Specification Document
     - Manufacturer Environmental Testing and Total Running Time
     - Delivery Date

- Waivers
- Functional/Aliveness Bench Tests
- Electrical Integration Procedure
- Performance Tests
2. Spacecraft Level Testing
    - Aliveness, Functional, and Phasing Tests
    - Comprehensive Performance Testing CPT)
    - Dynamic Simulator Testing
3. Is the Spacecraft Ready for Environmental Testing?

Pre-Ship Review (PSR)

A review is conducted before a spacecraft is shipped to its launch site to determine if it is ready for launch. This Pre-Ship Review would cover the following types of topics:

1. Events since PER
    - Spacecraft Tests w/Dates
        - Results
        - Completion Status
    - Flight Hardware Operating Hours
    - Anomalies Since PER
        - Problem
        - Cause of problem
        - Current status
2. Is the Spacecraft Ready for Launch?

CONCLUSION

This paper has attempted to cover a lot of material very broadly, to give an idea of what is involved in one approach to a successful spacecraft design. It is by no means the only approach to how to design a spacecraft, nor is this paper meant to be a detailed blueprint of the design process. It is, instead, a summary of some of the experiences that we have had at Goddard over the years, and is hopefully the first step in a larger effort to document the design expertise currently present in the Guidance, Navigation, and Control Center.

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge Robert Spagnuolo, Otilia Rodriguez-Alvarez, Debbie Henretty, Maureen Bartholomew, Nelson Rubin, and Martin Frederick, all members of Goddard's Guidance, Navigation, and Control Center, for their assistance in the preparation of this paper.

REFERENCES

1. G. E. Mosier and J. W. Croft, "XTE Linear Analysis Objectives Plan", Goddard Space Flight Center, Code 712, Internal Memo.

2. R. J. Richards, "An Introduction to Dynamics and Control", Longman Group Limited, London, 1979.

3. J. Bracken, J. San, J. D'Agostino, and K. Barnes, "TRMM ACS Algorithm Document", Goddard Space Flight Center, Code 712, Build 5.5, September 16, 1997.

4. F. Tubb, R. McEwen, J. Farazian, and A. Waddell, "MSTI-3 Spacecraft Attitude Control Software Development using Automatic Code Generation", Integrated Systems, Inc.