A New Satellite Attitude Control System

Wendy I. Sullivan
Michael A. Paluszek
Princeton Satellite Systems, Inc.

Walter K. Daniel
CTA Space Systems, Inc.

Abstract

This paper describes a new satellite attitude control system architecture, called the SPACECRAFT CONTROL SYSTEM. A simplified version of this control system is scheduled to fly in 1997 onboard Indostar®, a commercial geosynchronous communications satellite. The control system includes transfer orbit, acquisition and mission orbit modes. The software architecture permits easy modification and upgrades making this system applicable to any satellite.

The control system described in this paper includes many innovative features. Several are designed to reduce operator workload, such as automatic momentum unloading, a simplified moding structure and automatic mode transitions. Others reduce the cost of accommodating changes to the spacecraft or porting the control system to other spacecraft, such as the use of the simplex algorithm onboard to distribute the three-axis torque commands among selected thrusters. The entire control system is constructed to be highly modular, and this characteristic is reflected in the C flight code. The modular nature of the control system makes modifications and additions easy to accomplish, and results in highly maintainable flight code.

This paper will describe the SPACECRAFT CONTROL SYSTEM architecture and algorithms in detail, with emphasis on the new features mentioned above. An overview of the control system design environment and high fidelity simulation will be provided. Finally, simulation results from the Indostar test program will be presented, along with results from the SPACECRAFT CONTROL SYSTEM simulation.

Background

Traditionally, satellite control has been ground operator intensive, requiring that most control operations be initiated from the ground. As the industry gains experience in satellite operations, it is possible to design more automation into satellite control systems.

Control System Architecture

There are three distinct phases during which the attitude control system (ACS) must operate: transfer orbit, which begins when the satellite separates from the launch vehicle and during which the satellite must attain its mission orbit; acquisition, which is the process of locating the sun and the earth and aligning the satellite properly on station so its mission can be carried out; and mission orbit, during which the satellite must maintain its correct attitude and station. A block diagram of the entire control system architecture is shown in Figure 1. The block diagram shows the interaction between the sensors, control modules and the actuators. The processors for all of the sensors are listed on the left-hand side of the diagram: horizon sensors (HSA), sun sensors (SSA), gyros, earth sensors (ESA), and the momentum wheel assembly (MWA) tachometer. The processed output from the sensors is used for error computation and input to the control systems (shown in the center of the diagram): transfer orbit control, earth acquisition control, and mission orbit control. Then, the outputs of the control systems are distributed to the actuators (on the right-hand side of the diagram): the rocket engine assembly (REA) thrusters, the momentum wheel motor, the magnetic torquers, and the electric hydrazine thrusters (EHTs). The modularity of the system is enhanced by separating error computation, the control laws, and the control distribution.

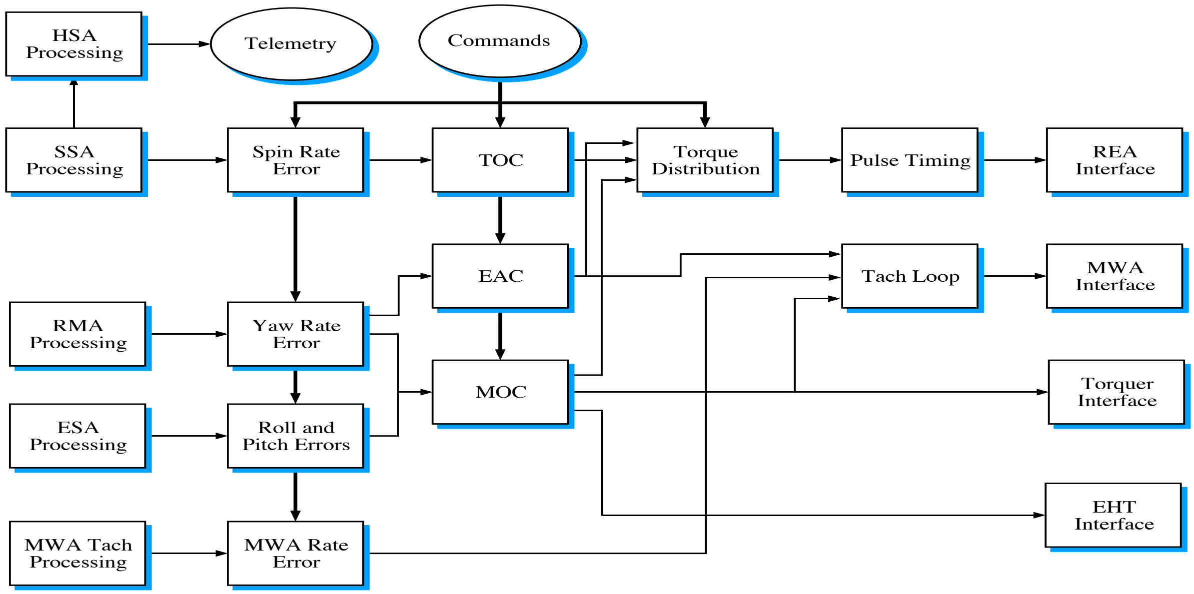

Figure 1. Control system architecture

The schematic shown in Figure 2 shows each of the ACS modes, and which mission phase they are used in, as well as

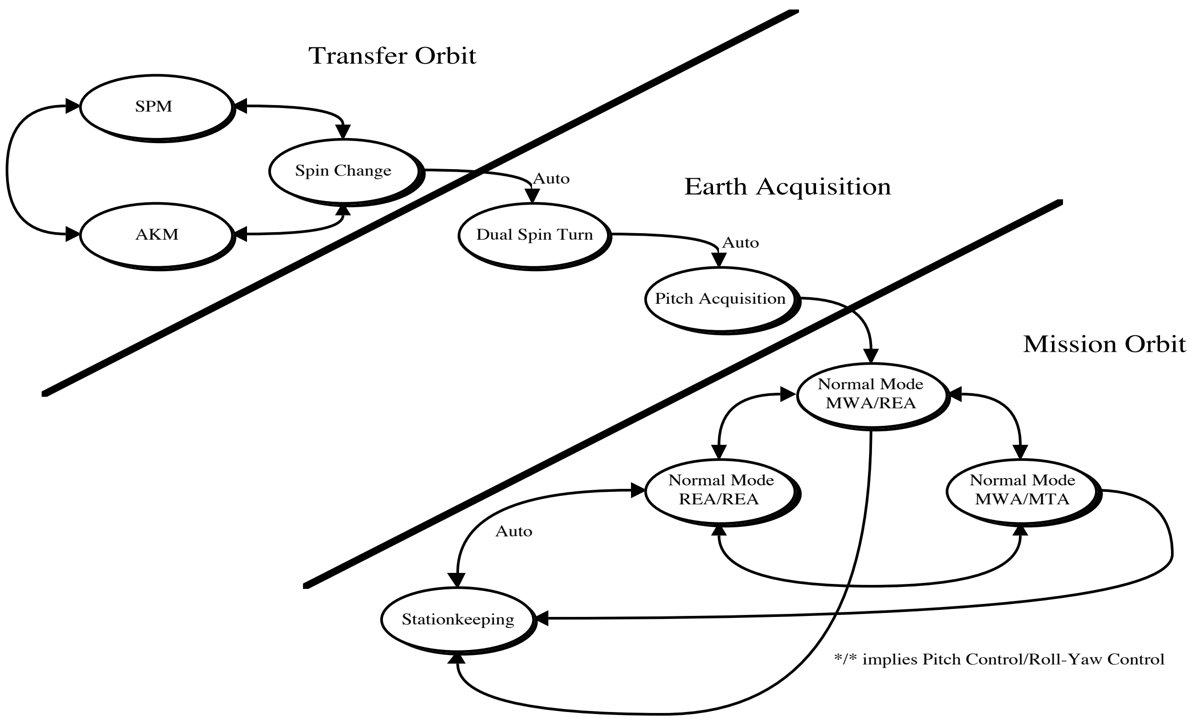

Figure 2. Moding diagram

the flow between modes. Notations indicate which mode transitions occur automatically; others must be initiated from the ground. This section will present the architecture in detail and describe each of the control loops in the system.

Transfer Orbit

During Transfer Orbit (TO) a solid rocket motor, called the apogee kick motor (AKM), is used for insertion into geosynchronous orbit. The satellite is spin stabilized during the rocket firing. During this phase all control functions and mode transitions are initiated from the ground. After separation from the launch vehicle, the satellite operator commands the satellite to spin about the AKM firing axis. A proportional control law is used for the spin-up maneuver, based on a spin rate measurement from the SSA pulses and a spin rate estimator structure. The estimator accounts for spin rate changes due to firing thrusters. The estimator equations are:

$$\begin{aligned} \omega_E &= \omega + \alpha p \\ \omega_M &= \frac{2\pi}{T} + \frac{\alpha p^2}{2T} \\ \omega &= \omega_E + K(\omega_M - \omega_E) \end{aligned} \tag{1}$$

where α is the nominal acceleration per unit time, p is the thruster pulsewidth, and T is the measured spin period.

The operator also has the option of manually choosing and firing thrusters to change the spin rate. After the spin change maneuver, it is necessary to align the spin axis with the desired direction of the rocket firing. The reorientation is done using a spin precession maneuver (SPM) with pre-calculated thruster firings based on the sun pulse timing. If separation is not done at the AKM firing attitude a large spin precession maneuver will be required and the large spin precession maneuver will be done prior to spinning the spacecraft up to its AKM spin rate.

During the AKM firing, the spacecraft is spin stabilized, and no active control is needed. After the firing is completed, another spin change is needed to prepare for the acquisition sequence.

Earth Acquisition (EA)

During acquisition a dual-spin turn (DST) is used rotate the satellite into the correct attitude, and the resulting pitch rotation is damped to minus orbit rate. The control law during the DST is open loop, in that the maneuver is accomplished simply by commanding a MWA speed. The tachometer loop which controls the wheel speed is described in more detail in the mission orbit section.

After the DST, the spacecraft transitions to pitch acquisition mode.The pitch acquisition system is a simple proportional derivative controller using the thrusters. This pitch controller is also used during mission orbit and the gains are set prior to acquisition to emphasize rate damping as opposed to position control.

During acquisition, the spacecraft changes from spin stabilization to bias momentum three-axis control.

Mission Orbit (MO)

During the mission orbit phase the satellite uses a bias momentum design. Changing the speed of the momentum wheel controls pitch, and orthogonal magnetic torquers are used to control roll & yaw. The earth sensor measures roll and pitch; yaw is not directly sensed, but can be determined from the orbit rate coupling with roll. A yaw gyro is used to sense rate information during stationkeeping maneuvers. EHTs are used for north-south stationkeeping maneuvers, and REA thrusters are used for east-west stationkeeping control, for momentum wheel momentum unloading and for backup roll/yaw control. REAs can also be used as a backup to the EHTs for north/south stationkeeping. EHTs cannot be pulsed and are not used for attitude control.

The rest of this section will describe the attitude control loops in detail and provide insight into the thruster operations.

The control system has several modes. They are listed in Table 1.

There are five controllers that can be used during MO: two normal mode pitch controllers, a normal mode roll/yaw controller, and the stationkeeping controller. The normal mode pitch controllers both apply proportional derivative

Table 1: Modes

Mode (pitch/transverse)	Description	Controller Type (pitch/transverse)	Actuators
MWA	Pitch control using the MWA only	PI	MWA
MWAMTA	Normal 3-axis control	PI / Low bandwidth (phase stabilized)	MWA - pitch; MTA - roll/yaw
MWAREA	Pitch control using the MWA and roll/yaw control using thrusters	PI / Low bandwidth (phase stabilized)	MWA - pitch; REA - roll/yaw
REAREA	Three axis thruster control	PD/Low bandwidth (phase stabilized)	REA
Stationkeeping	Three axis thruster control	PID/PID (High bandwidth)	REA

(PD) control to the pitch axis; the MWA pitch loop is a proportional integral (PI) controller which controls the wheel speed demand and the REA pitch controller is a PD with control applied directly to the pitch axis. The normal mode roll/yaw controller is a low bandwidth controller which uses the roll/yaw coupling to control yaw. This controller structure is used for both the magnetic roll/yaw control system and thruster backup. During stationkeeping, all three axes are controlled by a high bandwidth proportional integral derivative (PID) controller.

Pitch Control. The basis for the pitch control using the MWA is a tachometer loop that maintains the speed of the momentum wheel. It is desirable to keep the wheel speed within a range of the nominal speed, normally ±10%, to keep the spacecraft gyroscopically stiff. A DC motor has the transfer function

$$\frac{\omega}{T} = \frac{1}{Js + \beta} \tag{2}$$

where β is due to the back emf and viscous friction in the motor. The simple control scheme used is to multiply the difference between the desired speed and the measured speed by a gain K and to filter the measured speed by a first order filter. The resulting closed loop system is

$$\Omega = \frac{(s + \omega_T)(K\Omega_C + T_C)}{Js^2 + (J\omega_T + \beta)s + \omega_T(K + \beta)} \tag{3}$$

where ω_T is the filter cutoff. ω_T was chosen to provide adequate damping and K is made sufficiently large to provide good disturbance rejection and command tracking.

The input to the tach loop is the wheel speed demand. The equation for the pitch loop is

$$I\ddot{\theta} + J\dot{\Omega} = T \tag{4}$$

so it follows that the wheel speed demand will be

$$\Omega = \frac{c\theta + k\int\theta}{J} \tag{5}$$

which is a proportional-integral controller. Since the term in the pitch body rate equation contains the derivative of Ω, this controller provides proportional derivative control action in pitch.

Normal mode pitch control may also be implemented using thrusters. In this case, a PD controller is used. The PD controller is of the form

$$T = K_P + K_R\frac{\omega_R s}{s + \omega_R} \tag{6}$$

The derivative term is multiplied by a first order filter to prevent differentiation at high frequencies. The gains are determined by selecting the closed loop pole locations. When selecting the pole locations, it is important to consider the phase lag introduced by the ESA noise filter.

During stationkeeping, pitch is controlled using the same high bandwidth controller as the roll and yaw axes.

Low Bandwidth Roll/Yaw Control. The roll yaw control system must accomplish two requirements. First, it must attenuate the external disturbances on the spacecraft. These are at harmonics of the orbit rate and generally have no significant components above twice orbit rate. The second requirement is nutation damping. Since there is no passive source of nutation damping, the control system must damp the nutation. Since the nutation frequency is much higher than orbit rate the controller is broken into two parts, one to attenuate low frequency disturbances and the second to damp nutation.

The first part of the controller takes the low frequency approximation to the open loop system and selects a pair of gains to meet the pointing requirements. This approach, a purely proportional control, is the simplest. The nominal plant of an earth pointing momentum bias spacecraft has poles at orbit rate and at the nutation frequency. Since the nutation poles are undamped adding the low frequency compensation will destabilize the nutation poles. Consequently, it is necessary to phase stabilize the nutation mode. A second order compensator is used to stabilize the nutation mode.

The low-frequency approximations of the roll yaw equations are

$$\begin{aligned} -\frac{T_z}{h_w} &= \dot{\theta}_x - \omega_o\theta_z \\ \frac{T_x}{h_w} &= \dot{\theta}_z + \omega_o\theta_x \end{aligned} \tag{7}$$

and the torque command is

$$\begin{bmatrix} T_x \\ T_z \end{bmatrix} = h_w \begin{bmatrix} K_{xx} \\ K_{zx} \end{bmatrix} \theta_x \tag{8}$$

since only roll is measured. This control system is also used for the normal thruster control modes.

Nutation damping is accomplished by introducing a complex zero just before the nutation pole pair using the transfer function in (9) and a damped pole after the nutation mode.

$$\frac{s^2 + \omega_z^2}{s^2 + 2\zeta\omega_p + \omega_p^2} \tag{9}$$

As long as the zero is at a frequency less than that of the nutation pole pair the system will be stable. The compensator Bode plot is shown in Figure 3

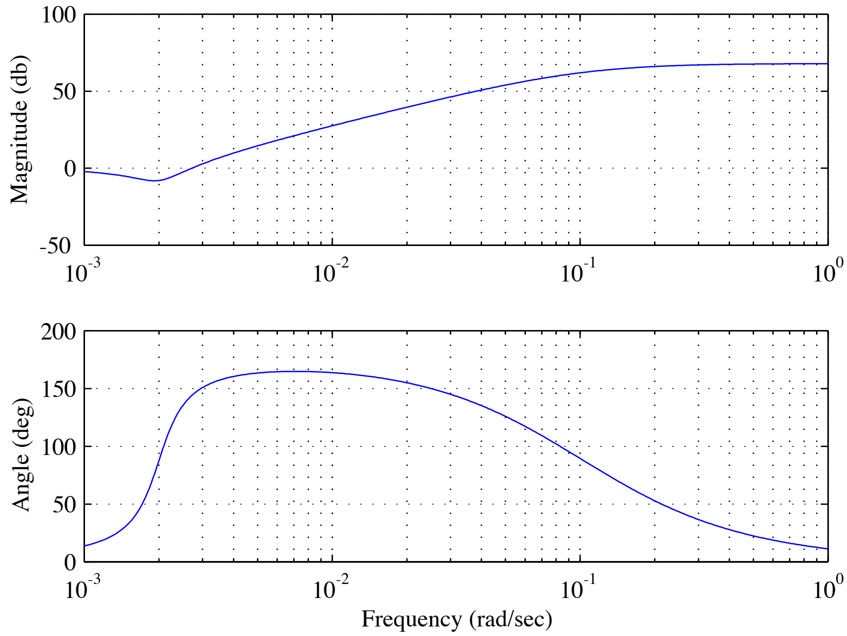

Figure 3. Nutation compensator Bode plot

The system with compensation is shown below.

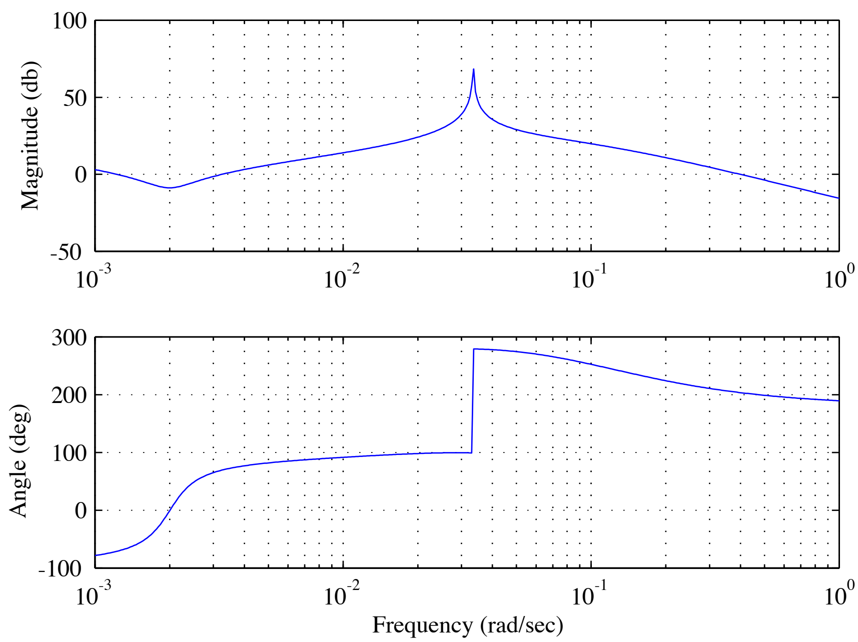

Figure 4. Compensated system

High Bandwidth Control. In high-bandwidth mode, each axis is controlled independently with a proportional-integral-differential controller. The yaw measurement is taken from the yaw gyro. The yaw gyro does not provide an absolute attitude reference and the control system assumes that the initial yaw attitude is zero. This does not pose a problem as long as the mode is not run for long periods of time.

It is assumed that the rates and attitude errors are small, and then the axes are decoupled and the control problem is reduced to

$$u = \ddot{\theta} \tag{10}$$

or three decoupled double integrators. These equations are valid if the flex modes are at much higher frequencies than the control bandwidth and, in our case, the bandwidth is much higher than the nutation mode. If this is the case then each axis can have an independent control loop.

The PID controller is of the form

$$T = K_P + K_R \frac{\omega_R s}{s + \omega_R} + \frac{K_I}{s} \tag{11}$$

The derivative term is multiplied by a first order filter to prevent differentiation at high frequencies.

In this system the plant is not a set of pure double integrators, instead it has the undamped nutation pole. Therefore, the bandwidth is set high enough so that the nutation pole is unobservable to the control system. This does not mean it has disappeared and the nutation mode will be excited by control activity and the disturbance torques. The normal mode thruster control is used to damp the nutation after the PID is turned off.

Thruster Operations. During thruster operations the 3-axis torque commands generated by the controllers are fed into the simplex algorithm, along with the positions and thrust vectors of the available thrusters. The simplex linear programming algorithm is used to determine the optimal set of pulsewidth commands. Optimal is defined as using the minimum amount of fuel necessary to produce the requested torque. The use of simplex allows the operator to account for thruster misalignments, plume disturbances and center-of-mass motion by changing ground loadable parameters without any reprogramming. The simplex implementation is customized for this application making it efficient enough for use with a relatively slow flight computer. The torque distribution law automatically limits the number of iterations in simplex and tests each pulsewidth command for validity.

All of the thruster control loops use thruster pulsewidth modulation. The minimum pulsewidth is relatively large which can lead to limit cycling if the disturbances are small (which is particularly true during non stationkeeping operation). The control system allows the operator to choose a pulsing period that is longer than the control period. For example, the stationkeeping loops run at 2 Hz but a typical pulsing period will be 8 seconds, meaning that thrusters will only fire once every 16 control cycles. This reduces limit cycling significantly.

Design Environment

The primary tool in the design process for this attitude control system was Princeton Satellite Systems' SPACECRAFT CONTROL TOOLBOX. The toolbox contains approximately 500 functions which support all phases of the design, from the underlying mathematics to preparation of pointing and fuel budgets.

Prior to any work being done, a MATLAB database was created for all the design parameters to ensure consistency and allow for easy updating of parameters. The next step was to design the control algorithms using MATLAB and the SPACECRAFT CONTROL TOOLBOX. Preliminary simulations were also done in MATLAB to verify the design.

Next, the control algorithms were ported to C, and the mode interfaces added. A high fidelity simulation was written in C including detailed models of the dynamics, hardware and interfaces. The simulation produces MATLAB plot files so the results can be more easily interpreted. Verification testing was performed using the C simulation.

The schematic in Figure 5 shows the design flow. The majority of the design work was done in MATLAB; only the performance verification and the simulation user interface design were done outside MATLAB.

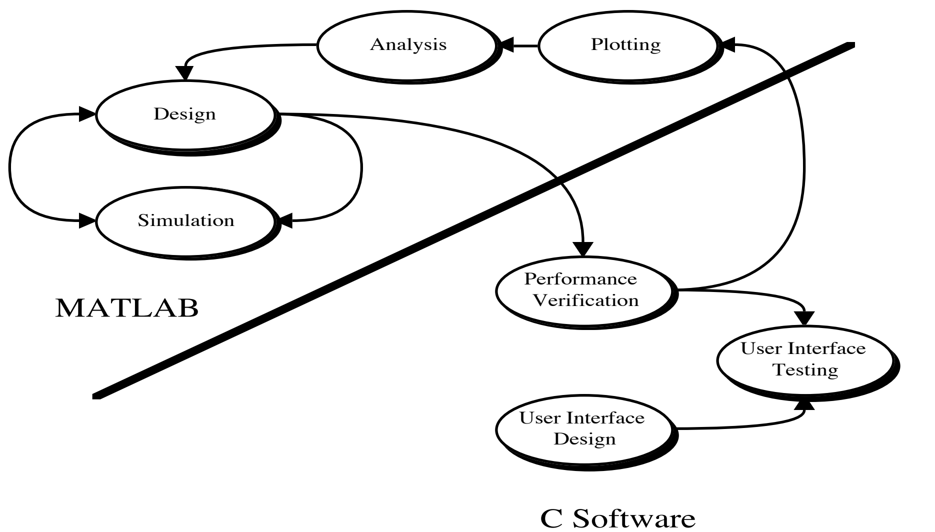

Figure 5. Design flow

Simulation

This section will discuss the simulation which was used to test the control system presented in this paper, including the models and user interface. The version of the SPACECRAFT CONTROL SYSTEM used and modified by CTA to test the Indostar flight software, INDOSIM, will be also be addressed.

Models

The spacecraft is represented by a non-linear multi-body model with the solar arrays and momentum wheels modeled as separate bodies each with a single axis of rotation. The kinematical equations of the core with respect to the inertial frame are represented with quaternions. The sensors modeled are a Barnes earth sensor, Adcole sun sensor, and Barnes horizon sensor. Environmental models include a solar pressure model, a constant magnetic field model, and a spherical gravitational model.

The earth sensor model models each of the twelve individual detectors and computes the amount of radiation impinging on each detector. The output of the model is in the exact binary format as the hardware.

User Interface

There are five parts to the SPACECRAFT CONTROL SYSTEM user interface: the command list, the telemetry file, the input file, the output file, and the output specification. The command list and telemetry file allow the simulation to be "flown," or operated, in the same way the actual satellite will be flown. The user is permitted to send one command list at the start of each simulation. Commands are tagged with a time for execution; the simulation will automatically send and execute the commands at the appropriate times. It is possible to run multiple simulations without further user interaction. The telemetry file collects all the telemetry from the spacecraft.

The commands and telemetry deal only with the flight software. The spacecraft model itself has all of its parameters embedded internally. However, many of them can be changed via the simulation input file. At the beginning of each simulation the file is read in and designated parameters are changed. The input file also allows the operator to start the simulation with any initial spacecraft state desired.

An output specification file tells the simulation how often to write to the output file, and which values to write. Outputs are allowed from both the flight software and the simulation. All output goes into a single ASCII text file, which can be imported into MATLAB or any other plotting package for plotting.

INDOSIM

The purpose of INDOSIM is to verify the performance of the flight software and to check interfaces to the 1750A flight computer used on Indostar. To meet this goal, two types of simulations were performed. First, pure software simulations were done, which included simulations of all of the flight software. Then, closed loop attitude determination and control system (ADACS) testing was done.

The ADACS closed-loop tests primarily involved two computers: the flight processor and the dynamics/environment simulator. The two machines communicated over a 1553 bus. The Indostar flight computer is a 1750 space-rated 16-bit processor. This device executes the ADACS flight software that is compiled using a pre-ANSI C 1750 compiler.

The attitude dynamics, orbital dynamics, and environment portions of the INDOSIM program are run on a 90 Mhz Pentium under MS-DOS version 6.2. The INDOSIM package was originally developed by Princeton Satellite Systems on a Macintosh computer and later modified by CTA Space Systems as the ADACS design evolved. The simulator receives actuator commands from the 1750, propagates dynamics, and returns simulated sensor data via the 1553 bus.

Results

Three categories of results are presented in this section. First, results from the SPACECRAFT CONTROL SYSTEM will be shown to demonstrate thruster pulsewidth modulation. Then, results from INDOSIM will be presented and compared with test results of the flight processor. These cases will show a stationkeeping maneuver, the transient response of the nutation control using MTAs, and the pitch acquisition performance.

In the following example, the SPACECRAFT CONTROL SYSTEM simulation was started with an initial roll rate with thruster control of all three axes. The normal mode (low bandwidth) controller is in use. The rate plots (on the left) in

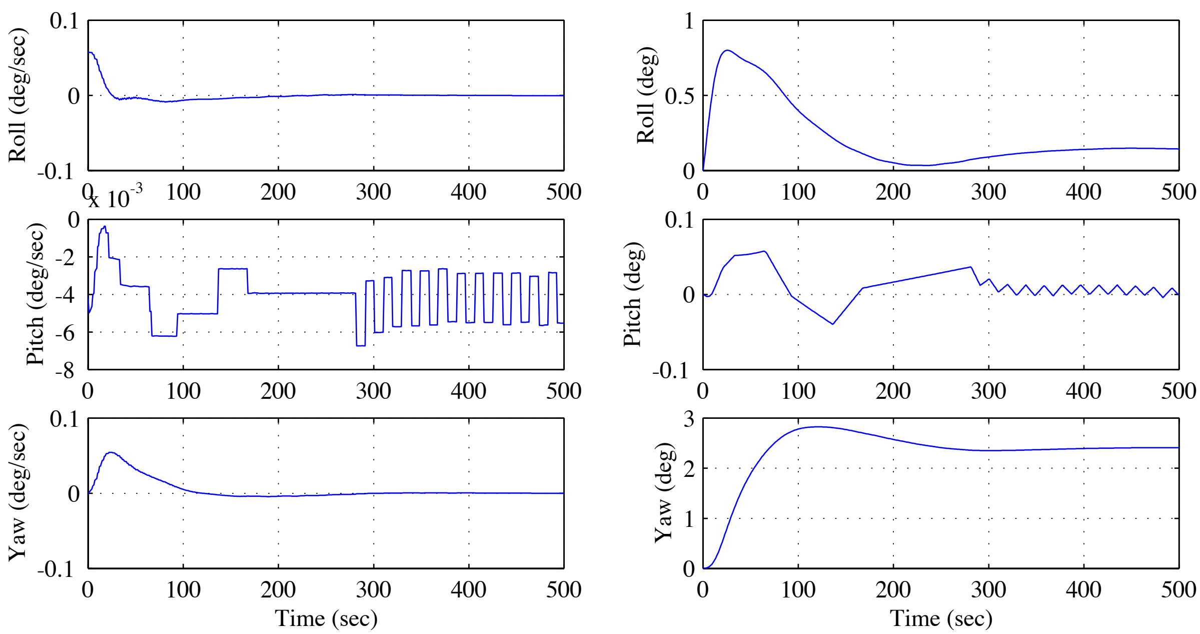

Figure 6. Thruster control

Figure 6 show nutation being damped; the attitude plots show the limit cycling in pitch. The thruster pulsing period was changed from 2 seconds to 8 seconds at 270 seconds. The trade-offs are demonstrated in the plot. Tighter control is achieved with the longer pulsing period, at the expense of more limit cycling and thruster firings. The plots of torque and torque demand shown in Figure 7 also show that with a shorter pulsing period it takes longer for the torque demand to reach the minimum pulsewidth, and therefore cause a thruster to fire. Notice that the torque is always delivered at the beginning of the control period. Care is required in choosing the pulsing period, because if it is too

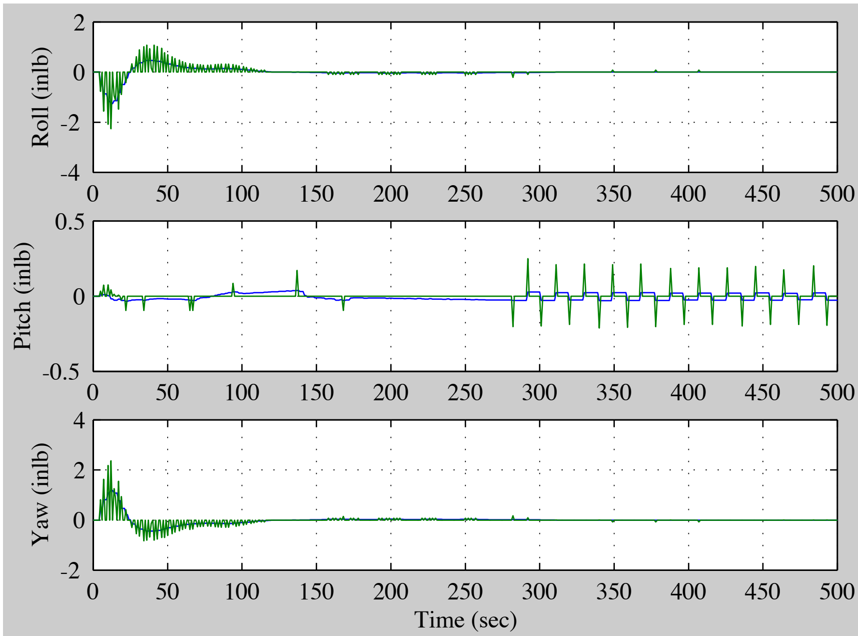

Figure 7. Torque and torque demand

long, and the delivered torque is not concentrated at the beginning of the period, then the control system will not work properly.

The next case shows a stationkeeping maneuver simulated completely within INDOSIM. The top plot in Figure 8

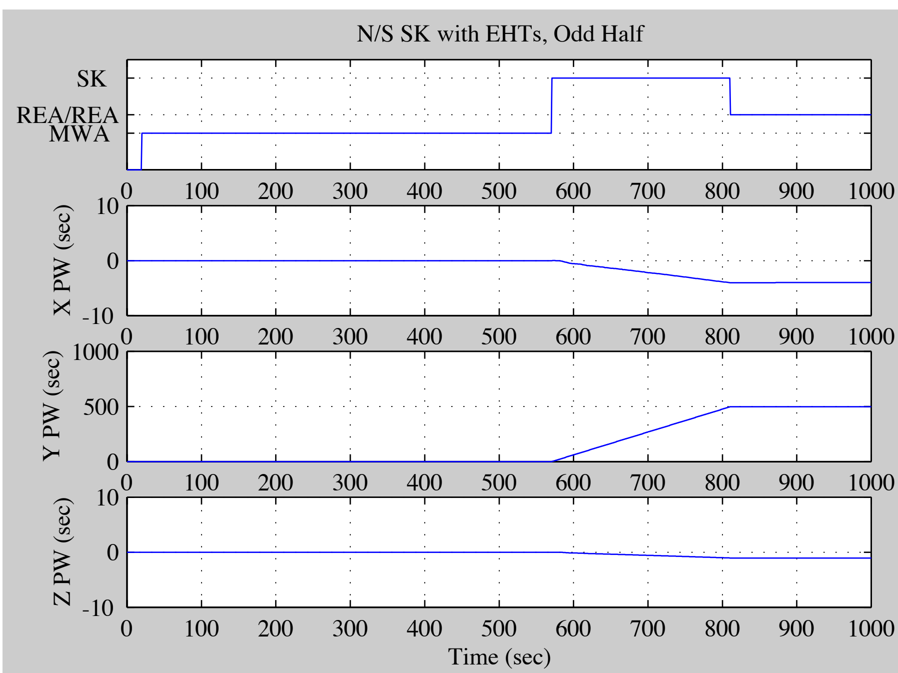

Figure 8. Stationkeeping modes and pulsewidths

shows the mode transitions, including the automatic transition from stationkeeping to REA/REA mode to enable nutation to be damped. Figure 9 shows the attitude during and after the maneuver, demonstrating that nutation was damped as soon as the maneuver ended. The limit cycling in pitch after the maneuver is due to the large minimum

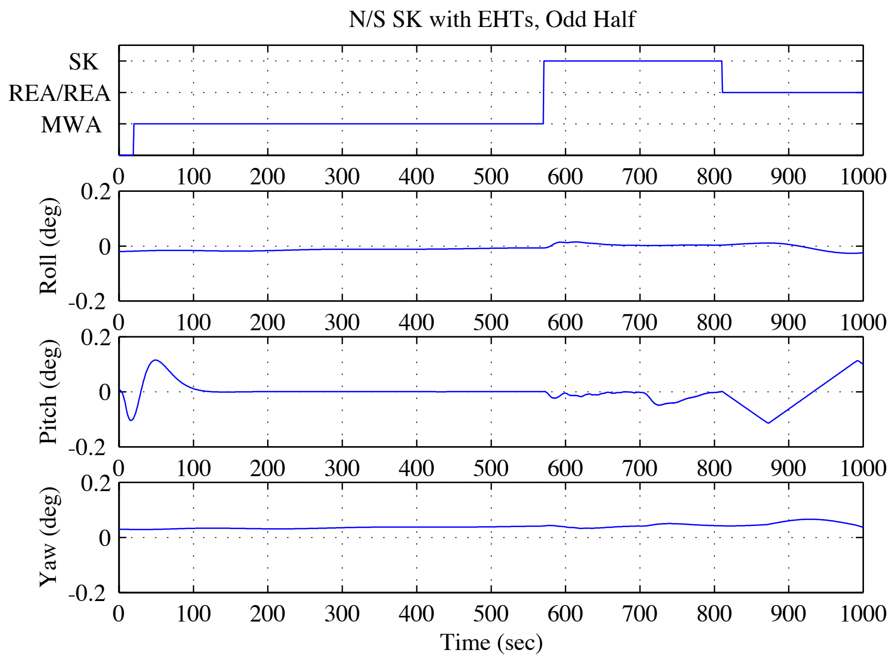

Figure 9. Attitude during stationkeeping maneuver

impulse of the thrusters relative to the pitch inertia of Indostar. A momentum dump was also executed during this stationkeeping maneuver, as shown in Figure 10.

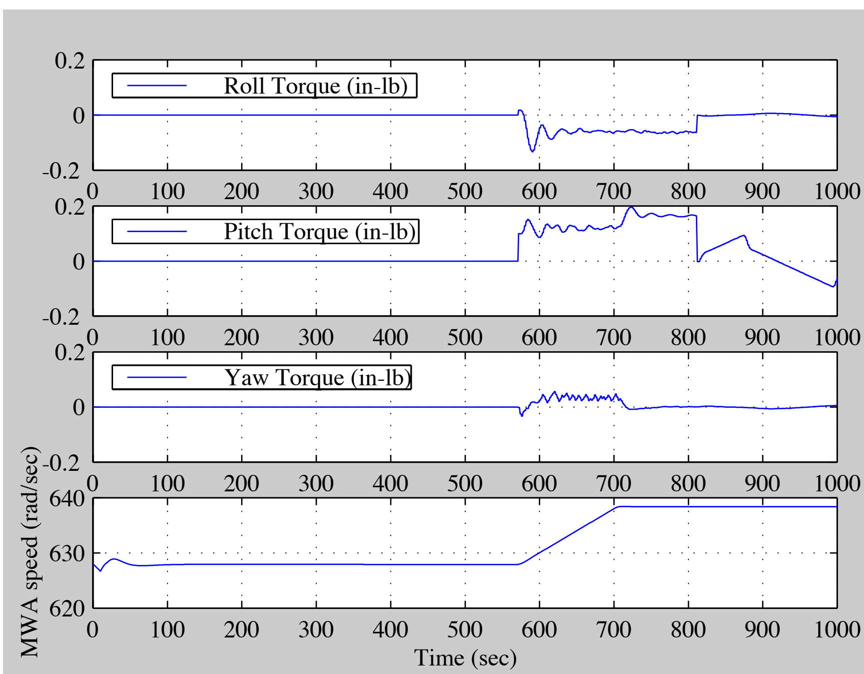

Figure 10. Momentum dump

The plots in Figure 11 show a pitch acquisition using thrusters simulated with INDOSIM. The spacecraft was given an initial rate of 0.2 RPM, and an initial angle of about 60˚. The acquisition occurs as soon as the earth passes into the field of view of the ESA. The limit cycling that was examined in the first example is also seen at the end of this run.

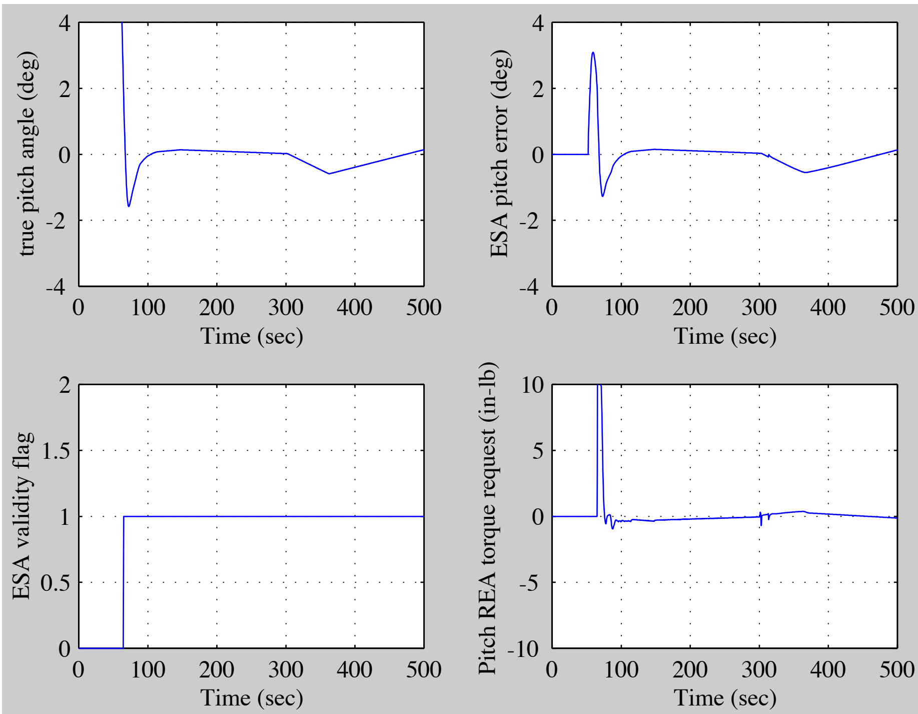

Figure 11. Indosim pitch acquisition

In the last case, the transient response is tested by manually firing a thruster to induce nutation, which is quickly damped by the MTAs. Figure 12 shows results from both the pure software version of INDOSIM and the version which used the 1750A to run the flight code. The results indicate that the control system is executing correctly on the flight processor.

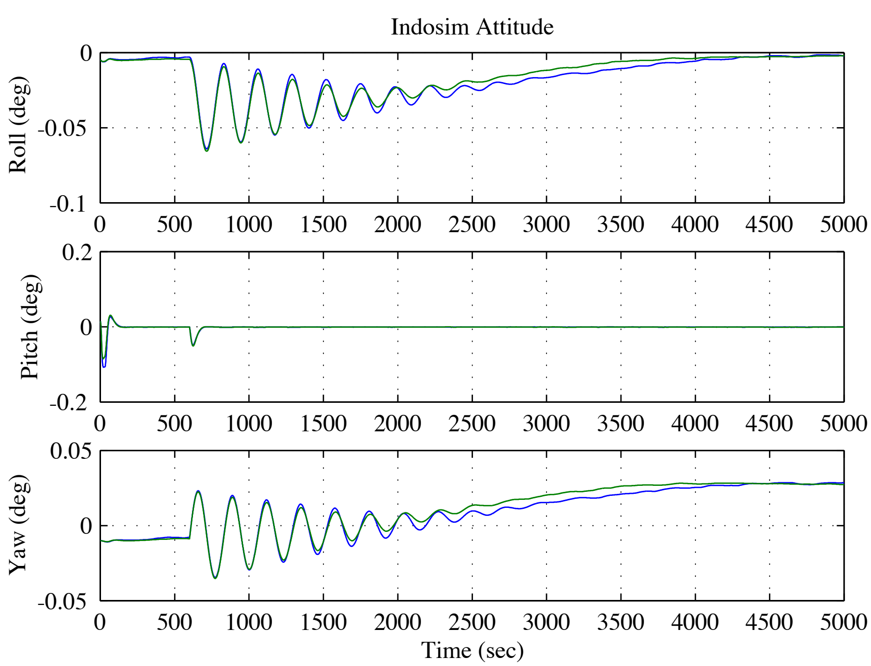

Figure 12. INDOSIM transient response- attitude

Conclusions and Suggestions for Future Work

In conclusion, a new satellite attitude control system has been designed which has several improvements over existing designs. These include the use of the simplex algorithm to distribute the torque commands and automatic damping of the nutation caused by stationkeeping maneuvers. Another advantage of this control system and the flight software which implements it is its modular design. The SPACECRAFT CONTROL SYSTEM allows for easy and realistic testing of the flight code.

Planned upgrades to the SPACECRAFT CONTROL SYSTEM include additional sensor models, an enhanced command language, and the creation of a simulation log file. The enhanced command language would allow loops and sublists of commands. The log file would record all major simulation events, such as commands, failures, and mode transitions. Another possible improvement to the simulation would be a graphical user interface (GUI) which allows the user to view the output data as the simulation is progressing. A GUI could also allow command lists to be sent at any time during the simulation, increasing the value of the simulation as an operator training tool.

Additional features of Version 2.0 of the attitude control system will include a complete gyro-based attitude determination system that can use earth sensor inputs, star sensor inputs or sun sensor inputs as measurements. A reaction wheel version will also be added with an advanced friction compensation system. An onboard spin-axis attitude determination system will also be ported from Princeton Satellite Systems' Spin Axis Attitude Determination Toolbox.