AttSim, Attitude Simulation with Control Software in the Loop

Hans J. Koenigsmann, Gwynne Gurevich

Microcosm, Inc.
2377 Crenshaw Blvd, Suite 350, Torrance, CA 90501
Tel: (310) 320-0555
E-mail: hansk@smad.com , ggurevich@smad.com
Website: www.smad.com

Abstract: AttSim is a spacecraft attitude simulator that has been specifically developed to design and verify attitude control concepts and flight software architectures and algorithms. Its primary goal is to provide a generic approach to small satellite attitude control development by allowing scalable performance. AttSim specifically allows the user to develop software modules that can be used as flight code, and to verify control logic, controller gains, and other mission-critical elements. The code can be developed in a flight-like environment, allowing quick conversion to actual flight code for a target processor. In addition, AttSim can be used as a systems engineering tool to ensure correct torquer, wheel, and thruster sizing and sensor performance. AttSim allows derivation of subsystem requirements to meet attitude accuracy (pointing and stability) goals. Its use during all phases can help reduce development and verification time as well as cost and risk.

Several missions have been developed to verify the AttSim system architecture and the cost savings associated with its implementation. The simulator's performance has also been verified using actual mission data, showing the simulations deliver realistic data. Using this accelerated, integrated development approach for the attitude control system, small satellites can be developed at lower cost and risk.

Introduction

Among all spacecraft subsystems, a large amount of resources is typically allocated to the attitude control system. This holds for design and development as well as integration, test and operations. Small satellites have a wide range of attitude control requirements, ranging from uncontrolled, passively controlled, to complex, momentum biased or reaction wheel systems. As a consequence, the cost of the attitude control system varies significantly. Sarsfield [1998] lists the cost of the "average NASA small satellite" attitude control system with 18% of the cost of the bus and at third place behind the launch cost and the operations. Wertz [1996] includes a cost break-down for 3 sample missions, with ACS representing 3-10% of the bus costs; these missions are, however, either spinner or gravity gradient attitude control systems, the least expensive ACS choices available.

Almost equally important, and related to the cost, is the risk of a failure caused by a total or partial failure of a subsystem. Statistics on the causes of failures are somewhat sparse, although two general trends have been reported (Sarsfield, 1998): 1st, the overall decrease in the number of spacecraft failures and, 2nd, the more significant role of design failures, as the total number of failures diminishes. The attitude control system is one of the subsystems with a tendency to have design failures, reversed signs in flight code, and other negative effects.

AttSim has been developed to speed the design and development of attitude control flight software and to test the code thoroughly at varying levels of design. The code can be tested while the control software is still in the design phase, with hardware models and, with minor modifications[1], as a hardware in the loop test. As a consequence, it allows more time for testing and, in most cases, a more realistic test, because it uses the flight code as controller with all non-linear and saturation effects, if modeled. This reduces the risk of design failures.

---

[1] Modifications include integration of a data acquisition card and the interface functions between the simulator and the data channels.

Attitude determination and control of small satellites

Small satellites have special requirements for attitude control systems, because power and mass are typically limited, but pointing requirements may, depending on the mission, still be tight. The simplest attitude control system is passive stabilization, either magnetically, aerodynamically, or by gravity gradient methods. Passive stabilization can be combined with active components, e.g. gravity gradient systems with magnetic torquers are quite common. Simple spinners, and momentum biased satellites represent the next advanced level of attitude control system, requiring at least for the momentum biased system some active stabilization about the angular momentum axis (typically the pitch axis). Zero-momentum systems with either reaction wheels or thrusters are the most complex systems, as they require constant stabilization and become unstable if control is lost only for a short period of time. Simple and passively stable systems have a low pointing performance, and complex systems using reaction wheels are highly accurate pointing systems.

Most small satellites in low earth orbit use magnetic attitude control with either an air coil or a magnetic torquer. Magnetic attitude control provides small and limited torque without consuming valuable fuel. The limitation of the torque is a directional limitation, because torque can only be created perpendicular to the Earth's magnetic field. The field reaches 60,000 nT at the poles and 30,000 nT at the equator, which results in $6*10^{-5}$ Nm per 1 Am^2 dipole moment (best case, if perpendicular to the earth magnetic field). The variation of the geomagnetic field with the satellite's position and attitude is one challenge during the control system design – minimizing the mass and power penalty of torquer or air coils is another. The size of the torquer has to be large enough to support both direct attitude control and angular momentum management.

One particular advantage of small satellites compared to larger spacecraft is the traditional lack of large, flexible appendices. At the same time, a truly rigid body provides little or no damping with respect to nutation, and either active nutation damping or a passive damper is required.

Other aspects of small satellite attitude control is flexibility, survivability, graceful degradation in case of failures and simple controllers. Secondary payload opportunities may leave the satellite in an undesired attitude after deployment, possibly rotating at high rates. Single ground stations have only limited contact, requiring a higher degree of autonomy than larger satellites in geostationary orbit.

Attitude Control System Design

There is, obviously, no single "correct" way to design an attitude control system, even if we attempt to formalize the process here from our point of view. As with other systems, the requirements are a good starting point.

Rarely are the control and knowledge requirements clearly stated numbers, such as 0.1 degree in pitch and 0.3 in roll and yaw, 3-sigma. They are usually derived from the payload requirements, and the needs of other systems such as power and TT&C. Other requirements on the ACS system are derived from such requirements as the maneuverability (*needs to turn 180 degrees in yaw in 2 minutes*) and the timing on the deployment sequences (*needs to orient the solar array to the sun within 30 minutes after deploy),* and thermal control requirements (*keep the sun off the battery panel*). These may be driving factors. Reliability and robustness, or the ability to continue controlling the spacecraft after a certain number of sensors and actuators failed must also be considered and will drive both the hardware selection and number of control modes required. The flight computer throughput, and the allowable mass and power for the subsystem are limiting factors, as well as the traditional driving requirement: cost.

When the requirements are defined, the type of ACS that would fulfill the requirements is chosen– such as momentum biased system, or a gravity gradient system. If possible, at least two options should be carried forward at this stage, to keep the cost and performance trade space open.

Sensor and actuator selection and specification configures the ACS and determines a major component of the system cost. At this stage, the design must also consider the modes required to meet the stated and derived requirements. This is the time when the requirements are re-visited, mostly in order to bring the cost down. Also at this time, a dynamical model is used to derive performance data or verify specific requirements are met, such as verifying that the solar array orientation can be obtained within 30 minutes. AttSim supports this design step. By using the AttSim controllers modified for the actual spacecraft if necessary, AttSim will verify the per-

2

formance of the controller and hardware combination. Performance of the different modes can be evaluated and trades made to arrive at a detailed design.

Flight code development and algorithm development may be separate tasks or combined as an integrated task, which has the potential of reducing the development cost. AttSim supports the integration, because algorithm modifications can be easily implemented and tested immediately, facilitating the flight code development process and turnaround time. Flight code and system development continues[2] culminating in a hardware-in-the-loop simulation, which could become a major system test, verifying other subsystems as well as the ACS. As mentioned above, AttSim has been used for hardware-in-the-loop tests, but requires system-specific modifications to interface with the analog to digital converter card that is being used.

Since most spacecraft have the capacity to upload flight software, ACS design may continue during operations. Unforeseen sensor and actuator failures can be simulated to support anomaly identification. Controller and filter updates and improvements can be tested on AttSim, before being uploaded to gain confidence in the newly configured system and its performance.

AttSim Overview

AttSim is an attitude simulator that has been developed for spacecraft attitude control design and verification. It specifically allows incremental software module development that can be used as flight code[3] and verifies control logic, controller gains and other mission critical elements. In addition, AttSim can be used as a system engineering tool to validate design concepts (correct torquer sizing, wheel sizing, sensor performance and related topics mentioned above). AttSim features sophisticated environmental models such as MSIS-86 (Hedin, 1988 and 1987) and the geomagnetic field model IGRF (Wertz, 1978, pp. 775), allowing a realistic simulation of atmospheric and magnetic disturbances. It sup-

ports a full moment of inertia matrix, three wheels (one per axis), and a structural damping factor, if appropriate.

AttSim runs on a IBM-compatible PC and is written in "C", compiled with Borland Turbo C 3.0. It runs under DOS or in a DOS window under Windows 95, 98 or NT[4]. AttSim can still be used as a hardware-in-the-loop simulator, but this function requires modifications to the hardware interface and is not part of the standard software package. The Borland C compiler produces stable code and allows easy and reliable debugging.

AttSim uses an input file with orbital parameters and the satellite initial attitude, and creates multiple output files plus a graphical file of the screen. The satellite parameters, such as mass, area, moments of inertia, are part of the program source code. AttSim currently has the following 5 controllers:

- Zero momentum, reaction wheel and torquer
- Zero Momentum with thruster
- Momentum biased, with earth, sun, or star sensor and gyrocompass
- Inertial scanner, momentum biased
- Gravity gradient with torquer

AttSim will run its own "flight control code" for each controller type, or can run customer supplied or newly developed control code.

An important feature of AttSim is the separation between the control module and the attitude simulation part of the program. This ensures that only data that are available in reality are used inside the controller module. There is one 'pipeline', or buffer, which transfers data from the simulator to the control module, and another 'pipeline' (buffer) that transfers data from the control module back to the simulator. Both are represented by arrows between the two parts in Figure 1.

The pipelines could, theoretically, be bypassed by global variables, but this would defeat the intent of AttSim, to be a flight code development tool. Figure 2 shows a typical AttSim screen

---

[2] Ignoring IV&V, because it is usually done by an independent organization. AttSim can be used on either side, but using it on both side would defeat the purpose of IV&V.

[3] After being adapted to the flight computer specific environment

---

[4] The real-time capacity, which was one of AttSim's original requirements as hardware-in-the-loop simulator, was easier to implement under DOS than under Windows 95 or 98.

output. There are five windows, with the two largest ones (windows 1 and 2) showing a unit sphere in inertial space. The satellite is in the center of this unit sphere, and its body axes are drawn onto the sphere to visualize the satellite's motion. The axes of the inertial coordinate system are also shown.
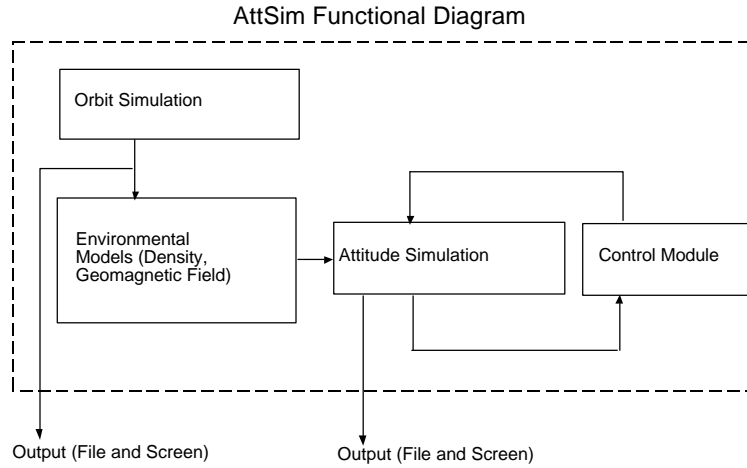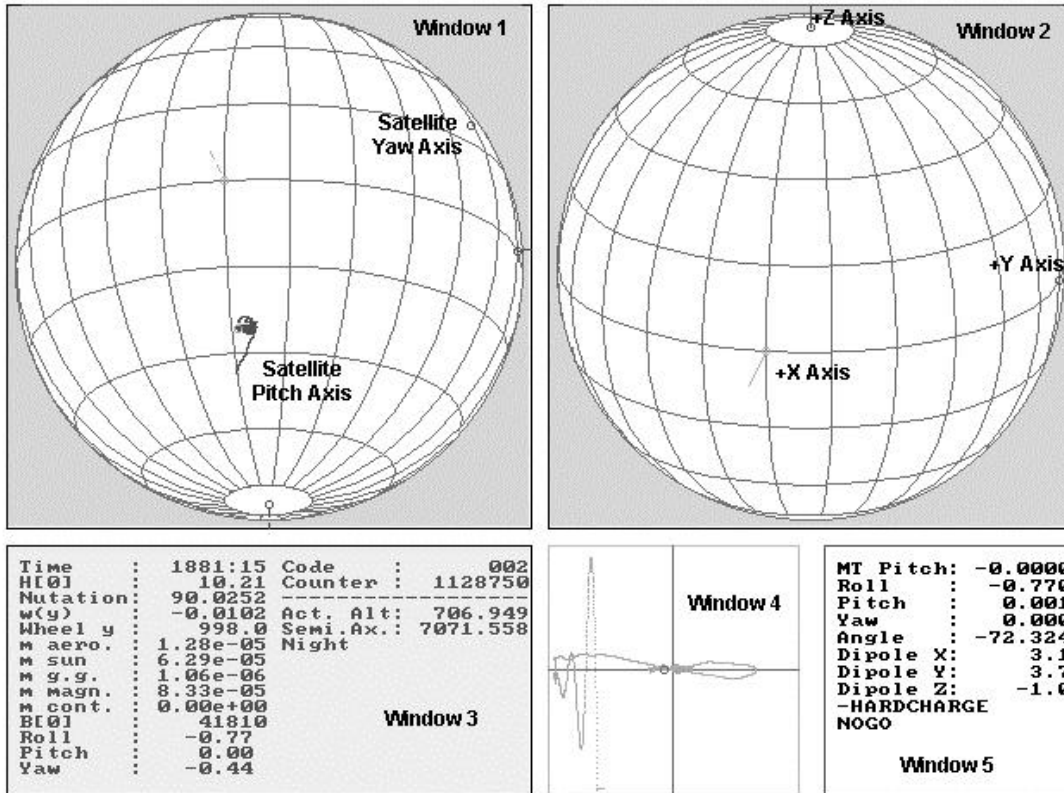
Figure 1: AttSim Block Diagram (Top-Level)

Figure 2: AttSim Screen Output

Only two axes of the spacecraft are usually shown on the unit-sphere, to avoid confusion. Typically, the Y (or pitch) axis is a "stiff" (mo- mentum biased) axis, and leaves a trace on the sphere. The other axis is the vehicle's X-axis, which, in case of an earth oriented spacecraft in a

4

circular orbit, points into flight direction and follows the earth horizon.

Window 3 shows simulation output, or "true" data – as opposed to window 5, showing controller telemetry data. The roll angles in window 3 and 5 differ due to sensor errors. Window 4 shows the Nadir angle in the vehicle coordinate frame, illustrating the attitude error in roll and pitch. The yaw angle can't be detected in this view, as it represents just a rotation about yaw. Note that, on the screen, the trace changes its color when the torquer is switched on, from blue to red.

Software Structure

AttSim basically consists of four loops running at different frequencies. The fastest loop determines the resolution with respect to time, which is usually set to 0.1 seconds. For a thruster implementation, this time step has to be set significantly lower, e.g. to 0.025 seconds or less, to detect minimum impulse bits. The minimum step size drives the attitude propagation, and updates the body matrix (transformation from inertial to body system). As shown in Figure 3, other functions are called at much lower frequency. The position, for example, is updated every 10 seconds, as are the environmental forces and torques, that change with it. The controller is usually called at 1 Hz; all time parameters can be modified and adapted to smaller or larger satellites.

Figure 3: AttSim Software Structure
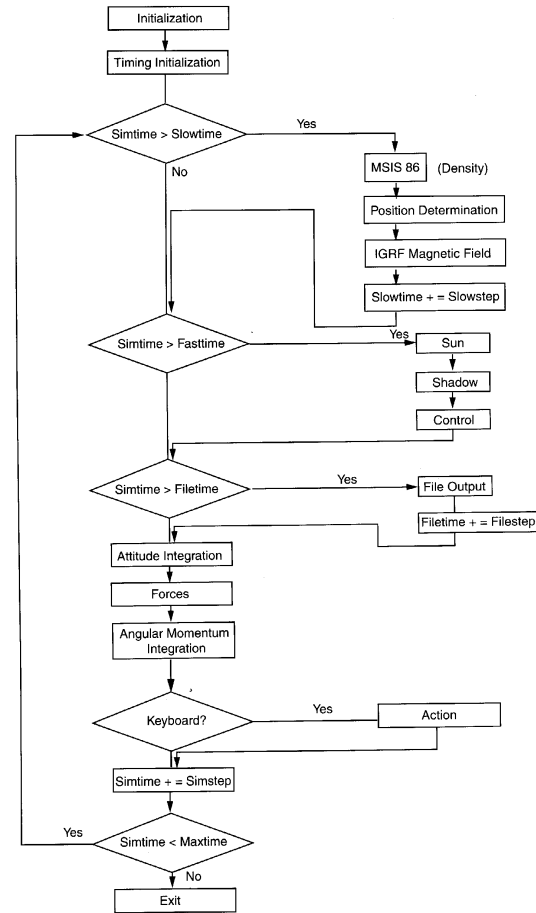
One of the key components of AttSim is the use of structured data, to combine easy access with comprehensive data. The satellite data structure contains all data that are required to propagate the state of a spacecraft, as shown in Table 1.

Table 1: AttSim Satellite Data Structure. ORBIT and VECTOR are additional data structures.

```
typedef struct
   {
   char name[MAXLINE];
   double tjd;          // True decimal julian date, incl. GMT
   ORBIT  *ce;          // Classical elements
   // Radius vector in varies coordinate systems
   VECTOR rr,rp,rq,rb,re,rs;
   // Velocity vector in varies coordinate systems
   VECTOR vr,vp,vq,vb,ve,vs;
   // Magnetic field vector in varies coordinate systems
   VECTOR br,bp,bq,bb,be,bs;
   // -----------------------------------------------------------
```

5

```
    VECTOR fa;           // Aerodynamic Forces (in inertial coordinates)
    VECTOR fs;           // Solar Pressure Forces ( " )
    VECTOR offa;         // Vector from CoM to CoP
    VECTOR offs;         // Vector from CoM to solar CoP
    VECTOR offd;         // Vector of magnetic Dipole moment of satellite
    VECTOR dm;           // Dipole moment of torquer, if any
    VECTOR mc;           // Control torque
    VECTOR ma;           // Aerodynamic torque
    VECTOR ms;           // Solar torque
    VECTOR mg;           // Gravity gradient torque
    VECTOR mb;           // Magnetic torque
    VECTOR m;            // Sum of all disturbance torques
    VECTOR w;            // Angular velocity in body coordinates
    VECTOR hw;           // Angular momentum of wheels....
    VECTOR ww;           // Wheel speed
    VECTOR mt;           // Motor torque of the wheel
    VECTOR h;            // Angular momentum of total sat. in body coordinates
    VECTOR q;            // Quaternion
    VECTOR euler;        // Euler angle, for better interpretation
    MATRIX I,W;          // Moment of inertia (I) and its inverse (W)
    MATRIX ab;           // Body Matrix

    double IW;           // Moment of inertia of wheel
    double mass;
    double area;
    double cd;
  } SATELLITE;
```

The file output frequency should be set to produce the amount of data desired. If the file output option is switched on, it produces six formatted ASCII files, which can be read by Matlab or other data analysis programs. The file output can be modified easily to accommodate data that are important to the user. For the generic file output, Matlab routines are available that produce graphs and data statistics. An example is given in Figure 4, showing the orbit of a satellite at 1000 km altitude with respect to a (fixed) Earth.

Other useful analysis tools include disturbance torque analysis, geomagnetic magnetic field analysis and eclipse analysis.

*Figure 4: AttSim Orbit Output, Converted With Matlab Routines.*
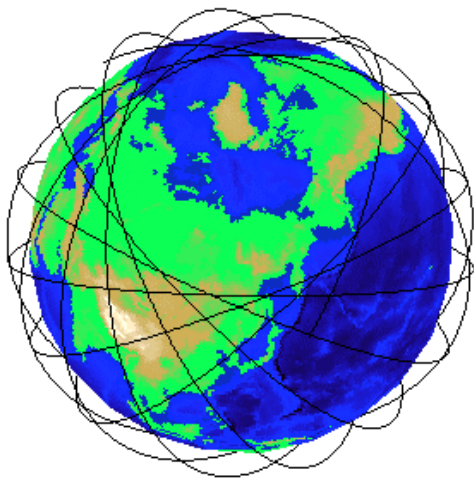
Control Code Implementation

The control module contains the "flight code" of the satellite attitude determination and control system. This module is called at a rate of 1 Hz (typically), and it uses a dedicated buffer to communicate with the simulator (the environment). The buffer is prepared in a separate function. The content of the control buffer is different depending on the type of controller being used; an earth oriented spacecraft needs other (simulated) sensor data than an inertial pointing spacecraft. The control module returns the sensor data it uses as telemetry, to allow comparison of the data created by the simulator and the sensor model data, which include noise. In general, the control module uses hardware interface functions to adds errors and bias to the sensor data, as shown in Figure 5. The (simulated) sensor data flow from the control buffer into sensor interface functions, which are called by the main function in the control module. The control module then calls a variety of other functions, depending on the control mode, such as a filter. The control module output – typically a momentum wheel torque or a dipole moment to the magnetic torquer – is processed by similar functions, the actuator interface functions. The actuator interface functions can add errors to the actuator commands, as it would be the case in the real world control system. Input and output share the control buffer.

Figure 5: Control Module Integration. Different controller may use different sensor data. Arrows to the right indicate subroutines that are being called from the controller.
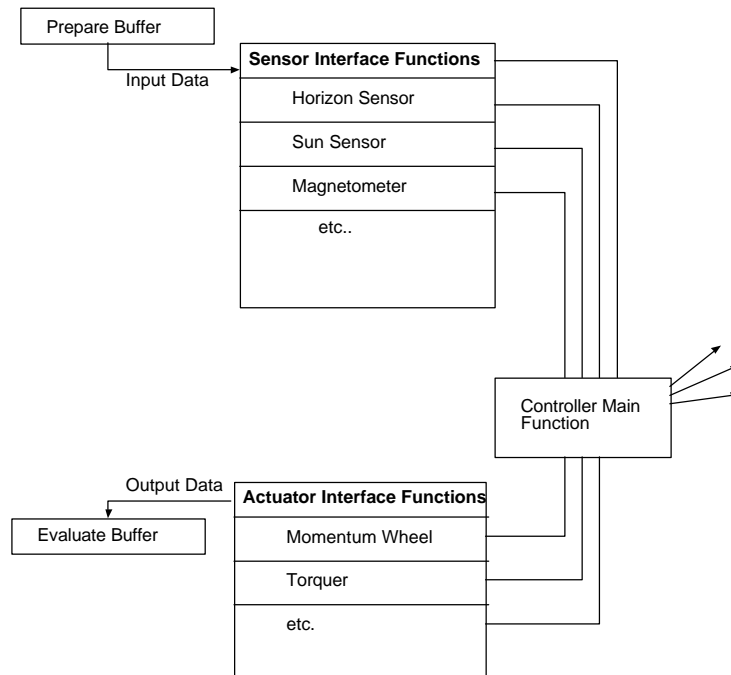
AttSim Implementation Example 1: Zero Momentum, Reaction Wheel Spacecraft

This sample satellite is a zero momentum spacecraft with three reaction wheels, two star sensors and three torquers. The spacecraft is an a circular, approximately 700 km altitude orbit with 56 degree inclination. Its mass is 500 kg, and the 10 m^2 array has, depending on the attitude angle, an offset of 1.5m to the center of mass. A residual dipole moment of 1 Am^2 in each axis provides additional disturbance torque, and the largest moment of inertia is in the pitch axis, and is likely to produce a constant gravity gradient torque. The input file is shown in Table 2. Note that most of the satellite parameters are defined in AttSim directly.

Table 2: AttSim Input File

```
Zero Momentum - Prototype
 2001          ; Year
    2          ; Month
    9          ; Day
   19          ; Hour
   23          ; Minute
   19.000000   ; Seconds
 7071.557960   ; Semimajor Axis A km
   33.083100   ; Mean Anomaly deg.
   56.952700   ; Incliniation deg.
     .000385   ; Eccentricity
   21.109100   ; Argument of perigee
  187.064900   ; RAAN, deg.
 5.0           ; Angle about Z0
-145.0         ; Angle about X1
0.0            ; Angle about Y2
```

At this altitude, the main disturbance torques (aerodynamic, solar, magnetic, and gravity gradient torque) are approximately of equal magnitudes. Using AttSim's output file, Figure 6 shows the magnitude of each torque. The solar torque drops to zero during eclipse phases, and aerodynamic torque varies according to the solar hour angle and due to eccentricity, and altitude variations. The atmospheric density, determined with a C-version of MSIS-86, varies by a factor of greater than 5 between the night and day side of the orbit. The residual magnetic torque is determined with the IGRF 10th order geomagnetic field model. The gravity gradient torque is, as expected, constant.

Roll, Pitch and Yaw error is shown in Figure 7. Initially, a large error is corrected, with significant overshoot in the pitch axis. Roll and Yaw axis then follow to some extent the aerodynamic torque.

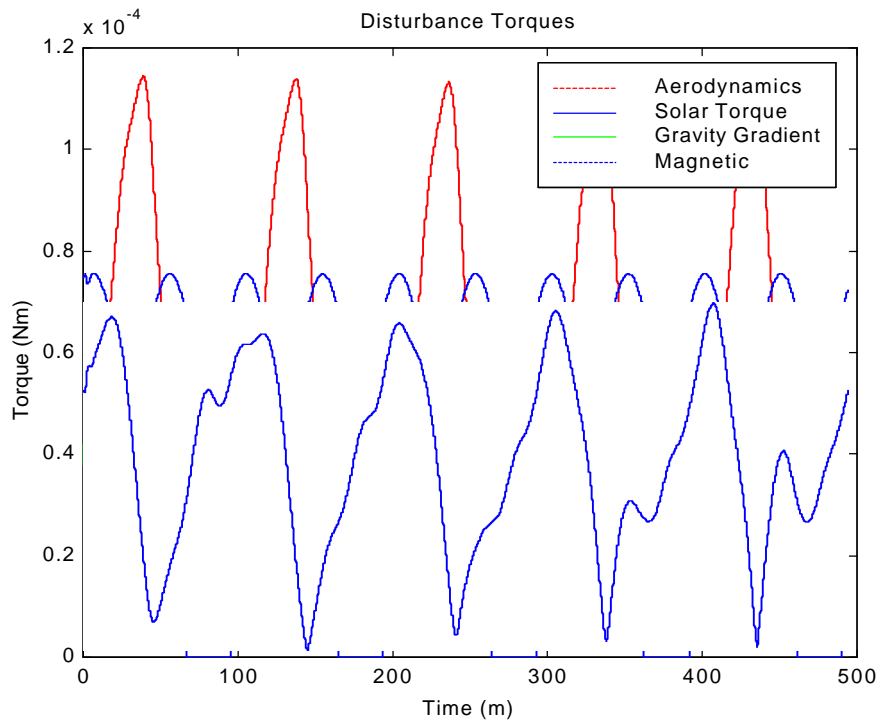*Figure 6: Disturbance Torque Analysis*
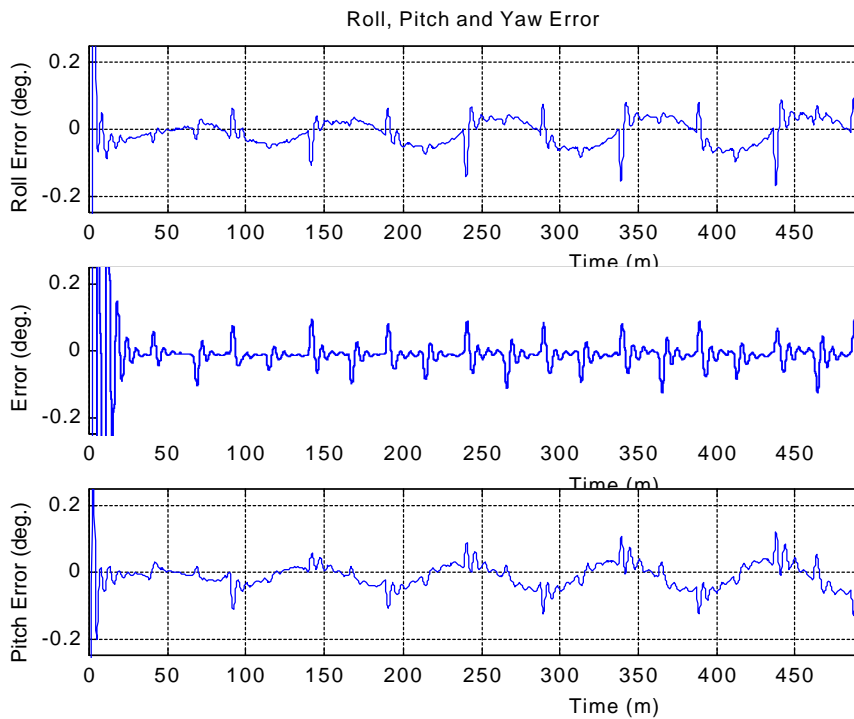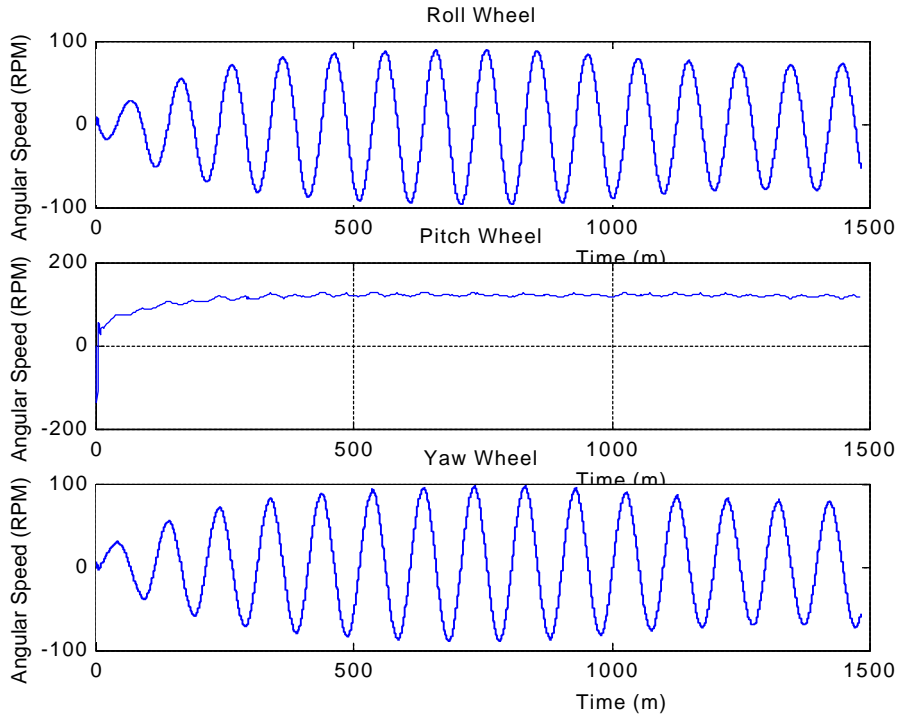

Figure 7: Attitude Errors

Koenigsmann   13th AIAA/USU Conference on Small Satellites

Figure 8: Momentum Management

Figure 9: Torquer Commands (Dipole moment)
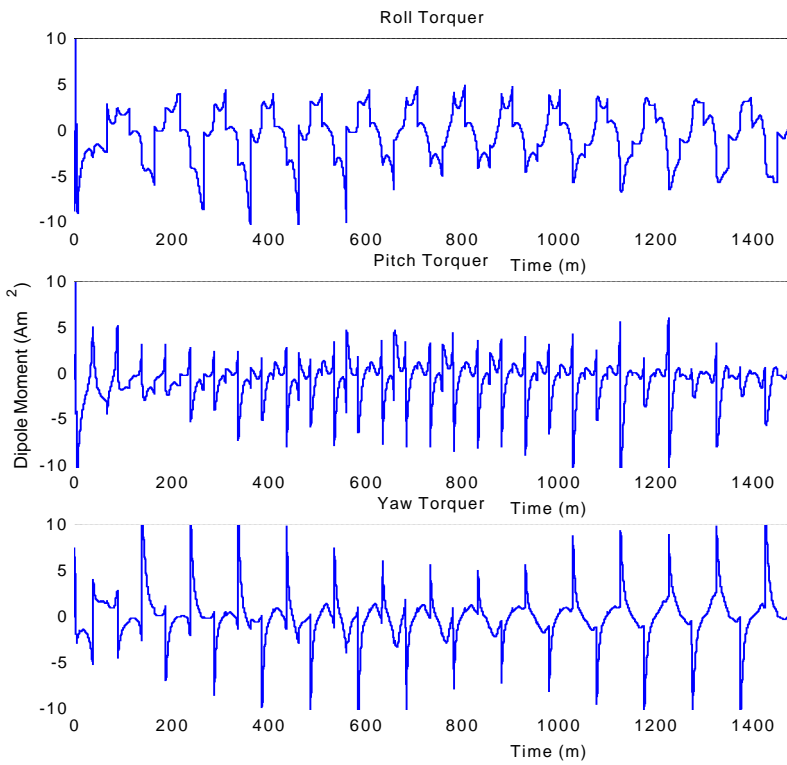
Koenigsmann   13th AIAA/USU Conference on Small Satellites

Although thermal flutter effects are not modeled, entry and exit in the earth shadow cause short term errors on every axis. The same controller is used on each axis, although the moments of inertia are different; Figure 7 suggests tuning the controller gains for each axis in order to improve the pointing accuracy. Using the Matlab analysis procedures developed for AttSim output files, statistical data are readily available: the pitch error standard deviation is just below 0.03 degrees, while roll and yaw errors are about 0.04 degrees.

Since this sample satellite is a zero momentum spacecraft, the reaction wheel speed is an important parameter. The wheel saturation routine drives the torquers constantly in order to keep the wheel speed within a certain range. The simulation over 5 orbits did not indicate stability, and a longer simulation was performed in order to evaluate the momentum management system. As shown in Figure 8, the wheel speed remains within a certain range, but the roll and yaw wheels constantly cross zero speed, which is undesirable.

An attempt was made to drive the wheels to an average rate of 100 RPM in roll and yaw, but resulted in instability due to the gyroscopic coupling, which the controller does not account for. A more advanced controller would be required to keep the wheel rates away from zero.

Nevertheless, this system is stable. A plot of the torquer commands shows that most of the time the torquer is switched to less than 10 Am^2 (Figure 9). For this mission mode, a torquer of that size would be sufficient. It also gives an indication of the duty cycle and allows to determine the (theoretical) average power consumption, if the simulation run is long enough.

Example 2: Gravity Gradient Satellite

This spacecraft uses the same orbit as the sample satellite above, but is a gravity gradient stabilized spacecraft with small aircoils, providing damping. The boom increases the moments of inertia for the roll and pitch axis to 160 and 180

kgm^2, respectively, while the yaw axis remains at 2.7 kgm^2. Unfortunately, the pitch rate does not match the orbital rate exactly, resulting in an initial error, that decays only very slowly. The dynamics model has no structural damping, and the aircoils only have 1 Am^2 dipole moment. The pitch axis trace on the unit sphere shows fairly large variations, and the motion resembles a pendulum (Figure 10). If, by using a simple "telecommand", the aircoils are deactivated, the situation becomes significantly worse, with the satellite stabilizing upside down (Figure 11). Note the trace of the body axis in Figure 11, that goes from the left unit sphere (the correct side) over the pole to the right unit sphere, the back side, and more and less stabilizes there. This is definitely not desired, and the small air coils prevent the spacecraft from doing this. Figure 10 still leaves doubts about the performance of this system, but the more detailed plot of roll, pitch and yaw, Figure 12, shows that roll and pitch remain within 2.5 degrees, while yaw has at the end of the simulation run a maximum error of 12 degrees. Structural damping has the potential to increase the performance of a stable system, and the assumption of zero structural damping is certainly a worst case assumption.
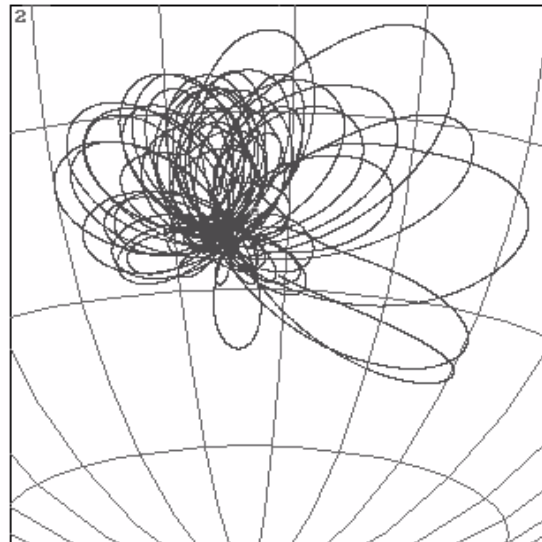
Figure 10: Gravity Gradient, Pitch Axis Motion
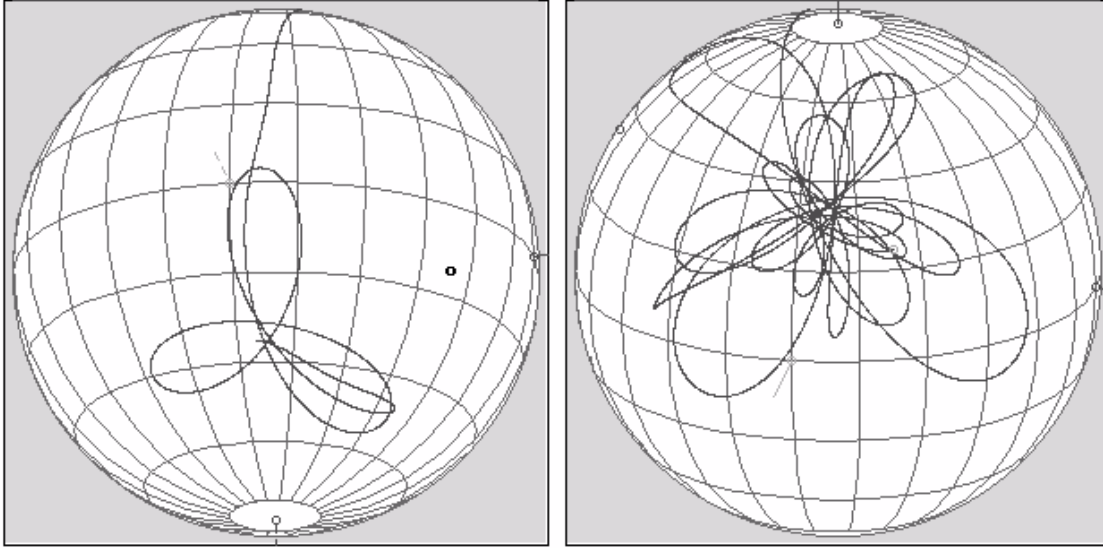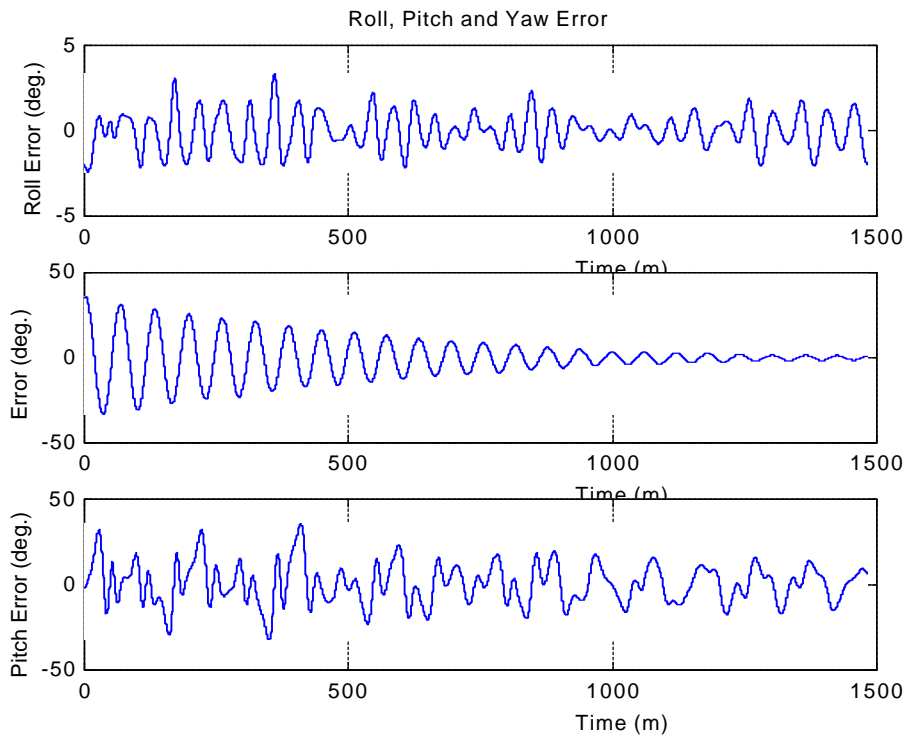
Figure 11: Gravity Gradient, Capture at Zenith

Figure 12: RPY with Air Coils

Koenigsmann   13th AIAA/USU Conference on Small Satellites

AttSim Verification

Attitude simulators are difficult to verify, because small initial errors may lead to larger, propagated errors with time. In addition, sign errors could occur in both the controller and simulator, and even the comparison with flight data, the best verification available, may not verify all functions. An interesting and long term comparison was made using TIROS-1 data from Bandeen [1960], Figure 13. For more than 30 days, the simulated spin vector tracks the measured vector very well. Since the spin rate of TIROS decayed throughout this period, a longer comparison would require modeling the spin rate decay in AttSim.

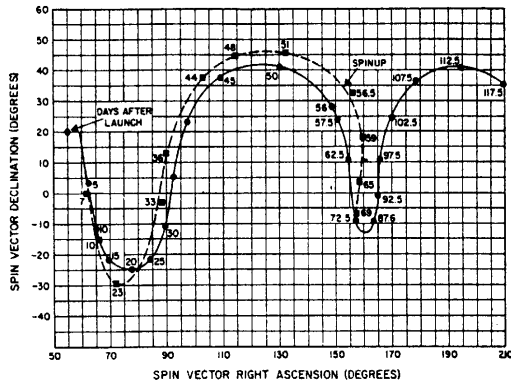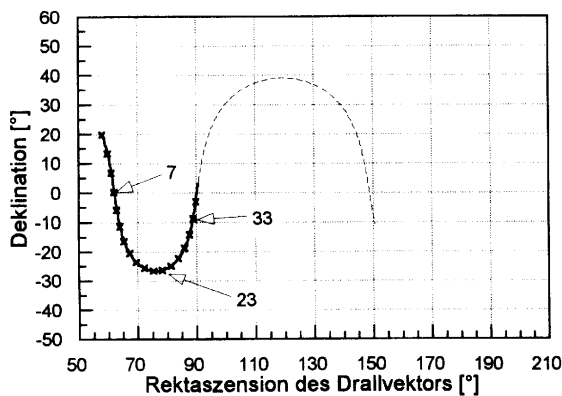*Figure 13 TIROS-1 Data from Bandeen [1960] and Simulated with AttSim*

Figure 14 AttSim Data Simulating the TIROS Spacecraft. Horizontal axis: right ascension of angular momentum, vertical axis: declination

Conclusions

The attitude control system development process can be a major cost and risk driver for any mission. AttSim is a versatile and powerful tool that integrates and simplifies the ACS design, development, test, and on-orbit operations processes. It allows detailed design analysis and facilitates investigation of second order and long-term non-linear affects. Even more importantly, it serves as a software-in-the-loop simulator. This feature integrates the development and testing tasks to allow incremental validation. Early test results are known to reduce the risk of failure for any system development. Finally, it serves (after adaptation) as a hardware-in-the-loop simulator and test platform for full system verification and on-orbit operations support tool.

References

Bandeen, William R. Manger, Warren P. ": Angular Motion of the Spin Axis of the TIROS 1 Meteorological Satellite Due to Magnetic and Gravitational Torques", *Journal Geophysical Research*, Vol. 65. No 9 (Letters to the editor), 1960

Liam Sarsfield, 1998: *The Cosmos on a Shoestring*, Rand

Wertz, James R. and Wiley, Larson J. (editors), 1996: Reducing Space Mission Cost, Microcosm Press, Torrance, and Kluwer Academic Publishers, Doordrecht

Wertz, James (ed.) : Spacecraft Attitude Determination and Control, 1978

Koenigsmann, Hans: "AttSim User Manual", Microcosm, 1999

Hughes, P. C. : *Spacecraft Attitude Dynamics*, J. Wiley & Sons, 1986

Francois Martel, et.al. "Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft", Utah State University Conference on Small Satellites, 1988

Hedin, Alan: "MSIS-86 Thermospheric Model", *Journal Geophysical Research*, Vol. 92, No. A5, May, 1987

Hedin, Alan: "The Atmospheric Model in the Region 90 to 2000 km", Adv. Space Res., Vol. 8, No. 5-6, 1988