Simulation, State Determination, and Control Design for a Small Imaging Spacecraft

David Friedman
Andrews Space
Seattle, WA 98168
770-826-9666
dfriedman@andrews-space.com

Eric Lund
Andrews Space
Seattle, WA 98168
206-438-0612
elund@andrews-space.com

***Abstract*—This paper presents the design of the Attitude Determination and Control System (ADCS) for the SCOUT imaging spacecraft being developed by Andrews Space. The unique performance constraints of low-cost space vehicles present control and estimation challenges in terms of processor throughput and sensor noise. We discuss the implementation of an ADCS able to achieve sub 100m accuracy (at 450km altitude while pointing 30° off nadir) by making use of steady-state Kalman filtering for attitude determination, low-order gravity models for orbit propagation, and an LQR-based control system. We discuss the impact of various noise sources on the satellites knowledge error as well as how processor-in-the-loop testing was performed to prove out the feasibility of running the ADCS system on the target embedded platform.**

TABLE OF CONTENTS

1 INTRODUCTION 1
2 SIMULATION TRUTH MODEL 1
3 ATTITUDE DETERMINATION 2
4 ORBIT PROPAGATION 4
5 CONTROL 4
6 WHEEL DE-SPIN LOGIC 6
7 PROCESSOR IN THE LOOP TESTING 7
8 SUMMARY 7
REFERENCES 8

1. INTRODUCTION

The SCOUT spacecraft is a low-cost imaging microsatellite being developed by Andrews Space with an expected initial launch date in mid 2015. In its stowed launch configuration, SCOUT has dimensions of 40 x 46 x 84 cm with a mass of approximately 50-55 kg depending on customer configuration. It can be launched in both a horizontal and vertical orientation as a secondary payload on a variety of launch vehicle platforms.

For attitude and position sensing, SCOUT is equipped with two PYXIS star trackers, three rate-integrating MEMS gyros, a three-axis magnetometer, six sun-sensing photodiodes, and a GPS receiver.

SCOUT's attitude control effectors are three 5 mN·m reaction wheels and three torque rods capable of creating a dipole moment of 15.5 A·m^2.

This paper provides a summary overview of SCOUT's attitude determination system, on-board orbit propagator, feedback control system, as well as the six degree-of-freedom (6 DoF) simulation (Figure 1) used to prove out the functionality of the guidance, navigation, and control (GNC) system. We also discuss the processor-in-the-loop (PIL) testing that was performed to ensure that the GNC system functioned given the processing and memory constraints of the satellite's embedded flight computer (PPC405 running Linux RTOS with 64 MB SDRAM).

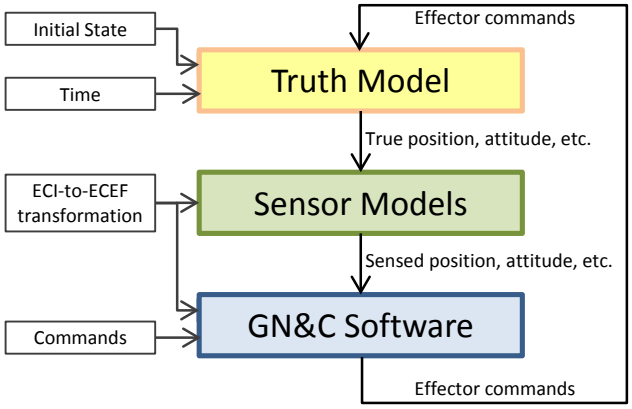

Figure 1. Top-level Simulation Overview.

2. SIMULATION TRUTH MODEL

A full 6-DoF GNC simulation was created in Mathworks Simulink® as this allows for simplified development, automatic generation of flight code, and easy integration with our processor-in-the-loop testing. This section covers each of the major components of the Truth Model. The core of the simulation has evolved at Andrews Space over the course of many years and many different projects.

ECI-to-ECEF Transformation

The ECI-to-ECEF transformation is necessary to convert back and forth between inertially-referenced vectors and Earth-Centered Earth-Fixed (ECEF) vectors. The math is directly based on that found in the USNO Circular 179.[1] However, due to processing constraints and the deterministic nature of the calculations, the transformation was precomputed for a range of dates. Within the simulation, a simple look-up table is used to obtain the instantaneous transformation quaternion. The accuracy of this method is <0.5 m for spacecraft position at 450 km.

Magnetic Field Model

The Earth magnetic field model is a direct Simulink® implementation of the International Geomagnetic Reference Field (IGRF-11) Model.[2] The general form is based on spheri-

978-1-4799-5380-6/15/$31.00

cal harmonics similar to the Earth Gravity Model described below.

$$V(r,\theta,\lambda,t) = R\sum\left(\frac{R}{r}\right)^{n+1}\sum_{m=0}^{n}\left(g_n^m(t)\cos m\lambda + h_n^m(t)\sin m\lambda\right)P_n^m(\theta)$$

Gravity Gradient Torque

The equations below are used to model the first-order gravity gradient torques on the spacecraft in body coordinates [3]

$$G_x = \frac{3\mu}{2R_0^3}(I_z - I_y)a_{23}a_{33}$$
$$G_y = \frac{3\mu}{2R_0^3}(I_z - I_x)a_{13}a_{33}$$
$$G_z = \frac{3\mu}{2R_0^3}(I_x - I_y)a_{13}a_{23}$$

where μ is Earth's gravitational parameter, R_0 is the radius of the orbit, the I terms are the spacecraft's principle moments of inertia, and a_{13}, a_{23}, and a_{33} are elements of the DCM derived from the body-to-inertial attitude quaternion.

Gravity Model

The gravity model within the Truth Model is based on the EGM96 Earth gravity model.[4] The same model is used within the orbit propagator of the GNC Flight code, but with fewer terms in order to speed up the calculation.

$$V = -\frac{\mu}{r} - \sum_{n=2}^{\infty}\sum_{m=0}^{n}\frac{\mu}{r}\left(\frac{\alpha_e}{r}\right)^n P_{n,m}(\epsilon)\left(C_{n,m}\cos m\lambda + S_{n,m}\sin m\lambda\right)$$

Aerodynamic Force and Moment

The aero force and moment follow from basic principles

$$F_{aero} = \frac{1}{2}\rho S_{REF} C_D \bar{V}^2$$
$$M_{aero} = \vec{CP} \times F_{aero}$$

where $\vec{CP}$ is the vector from the spacecraft's center of mass to its center of pressure. For the purposes of this simulation, Both the center of mass and center of pressure are assumed to be static with respect to the body frame. The $\vec{CP}$ vector is simply the vector from the center of mass to the center of pressure (in the body frame) projected into the local velocity frame.

Reaction Wheel Model

The reaction wheel model accepts an acceleration command and outputs the wheel's velocity. First the available torque is calculated as a function of the wheel's current velocity. It is assumed that motor torque falls off linearly with angular velocity.

$$\tau_{motor} = \tau_{stall} - \frac{\omega_{wheel}\tau_{stall}}{\omega_{wheel\_max}}$$

The acceleration commands are put through a transport delay of 50 ms and bounded by the available torque.

$$\dot{\omega}_{wheel} = \min\left(\dot{\omega}_{cmd}, \frac{\tau_{motor}}{I_{wheel}}\right)$$

The maximum wheel speed is software limited:

$$\omega_{wheel} = \min\left(\int \dot{\omega}_{wheel}\delta t, \omega_{wheel\_max}\right)$$

The motor constants ω_{wheel_max} and τ_{stall} were determined using a parameter ID process whereby the error between simulation and experimental data is minimized by tuning the parameters of interest.

Torque Rod Model

The torque rod model follows from basic principles

$$M_{torque\_rod} = (cD\hat{u}) \times \vec{B}$$

where c is the command (a scalar: $0 \leq c \leq 1$), D is the magnetic dipole of the torque rod at full power, and $\hat{u}$ is a unique vector that corresponds to the torque rods orientation in body coordinates.

Base Equations of Motion

The angular equations of motion were derived from Wertz [5]

$$\dot{\omega}_{sat} = \frac{\tau_b - \omega_{sat} \times [I_{sat}\omega_{sat} + I_{wheel}\omega_{wheel}] - I_{wheel}\dot{\omega}_{wheel}}{I_{sat}}$$

where τ_b is the sum of disturbance torques acting on the spacecraft including those from the magnetic torque rods.

The translation equations of motion are

$$\ddot{x}_{sat} = \ddot{x}_{gravity} + q_{ib}^*\left(\frac{F_b}{m_{sat}}\right)q_{ib}$$

where F_b is the sum of the forces in the body frame and q_{ib} is the inertial-to-body rotation.

3. ATTITUDE DETERMINATION

SCOUT's attitude determination system uses solutions from a star tracker, three-axis rate integrating gyro, and measurements of reaction wheel speed to estimate angular body rate. Attitude quaternions are not directly estimated (as in an MEFK approach) since the star tracker itself provides a low-noise quaternion estimate. Instead, the delayed quaternion from the star tracker is brought up to date using the angular body rate estimates.

The estimation process is broken into three steps which are covered in the following three subsections.

Angular Velocity Kalman Filter

A steady-state Kalman Filter is designed to estimate the satellite's body angular velocity as well as drifting biases in the gyro. The filter uses two measurements (gyro and star tracker quaternions) as well as one input (reaction wheel speed) per axis. Throughout this section, η_x is zero-mean gaussian noise with standard deviation σ_x.

The assumed gyro model is

$$\omega_{gyro} = \omega + b + \eta_{gyro}$$
$$\dot{b} = \eta_b$$

where b is the gyro bias.

Star tracker quaternions come in more slowly than the update of our GNC system (1 Hz vs 10 Hz) and have a typical delay of ~1 sec. With this in mind, we use the following approximate model to relate body rates to our periodic star tracker updates.

Using the relationship between body rate and quaternion-attitude:[6]

$$W(q) = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}$$

$$\omega = 2W(q)\dot{q}$$

Given the two most recent star tracker solutions q_k and q_{k-1} and the time between measurements Δt_{period}, we can approximate delayed body rate by:

$$\omega_{star} = 2W(q^k)\frac{[q^k - q^{k-1}]}{\Delta t_{period}} + \eta_{star}$$

ω_{star} can be thought of as the average body rate over the interval from k to $k-1$. It is assumed each measurement is delayed by Δt_{delay} meaning the average delay of ω_{star} will be:

$$\Delta t = \frac{2\Delta t_{delay} + \Delta t_{period}}{2}$$

Using a 1st order Padé approximate delay, a state-space model is built to approximate the relationship between the delayed measurement and true body angular velocity:

$$\dot{x}_d = \frac{-2}{\Delta t}x_d + \frac{4}{\Delta t}\omega$$
$$\omega_{star} = 2W(q_{k-1})\frac{[q^k - q^{k-1}]}{\Delta t_{period}} = x_d - \omega + \eta_{star}$$

The estimator also makes use of the measured wheel velocity, which is first numerically differenced, and then converted into satellite velocity using the previous estimate of body angular velocity:

$$\dot{\omega}_{wheel} \approx \frac{\omega^k_{wheel} - \omega^{k-1}_{wheel}}{\Delta t_{GNC}}$$
$$\dot{\omega}_{meas} \approx \frac{-\omega^{k-1} \times \left[I_{sat}\omega^{k-1} + I_{wheel}\omega_{wheel}\right] - I_{wheel}\dot{\omega}_{wheel}}{I_{sat}}$$

ω^{k-1} is the previous estimate of body angular velocity and the inertial terms, I_{sat} and I_{wheel}, are constants derived from CAD models of the satellite.

The complete system to be estimated is:

$$\begin{bmatrix} \dot{\omega} \\ \dot{x}_d \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ I_{3\times3}\frac{4}{\Delta t} & I_{3\times3}\frac{-2}{\Delta t} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \omega \\ x_d \\ b \end{bmatrix} + \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \end{bmatrix} \dot{\omega}_{meas}$$
$$\begin{bmatrix} \omega_{star} \\ \omega_{gyro} \end{bmatrix} = \begin{bmatrix} -I_{3\times3} & I_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} \omega \\ x_d \\ b \end{bmatrix}$$

which is a continuous time state-space model with the following state vector, input, output, A, B, and C matrices:

$$x = \begin{bmatrix} \omega \\ x_d \\ b \end{bmatrix}$$
$$u = \dot{\omega}_{meas}$$
$$y = \begin{bmatrix} \omega_{star} \\ \omega_{gyro} \end{bmatrix}$$
$$A = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ I_{3\times3}\frac{4}{\Delta t} & I_{3\times3}\frac{-2}{\Delta t} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}$$
$$B = \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \end{bmatrix}$$
$$C = \begin{bmatrix} -I_{3\times3} & I_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}$$

The measurement noise covariance matrix, R, is populated using noise statistics from the star tracker and gyros:

$$R = \begin{bmatrix} I_{3\times3}\sigma^2_{star} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3}\sigma^2_{gyro} \end{bmatrix}$$

The process noise covariance matrix, Q, is diagonal and tuned to achieve desired performance.

The gain, L, of a continuous time Kalman Filter at $t = \infty$ can be solved for by solving the algebraic Riccati equation:

$$AP + PA^T - PC^TR^{-1}CP + Q = 0$$
$$L = PC^TR^{-1}$$

This is used to build a continuous time state-space observer:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

Rearranging some terms:

$$\dot{\hat{x}} = (A - LC)\hat{x} + \begin{bmatrix} L & B \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix}$$

This is a standard state-space system with matrices:

$$A_{obs} = A - LC$$
$$B_{obs} = [L \quad B]$$
$$C_{obs} = I$$

This system is converted into a discrete-time state-space system using MATLAB's `c2d` function. Figure 2 shows the roll body rate estimate, truth, and the measured gyro data over a one minute interval.

Updating Stale Star Tracker Solutions

As previously mentioned, solutions from the star trackers are delayed by Δt_{delay}. For accurate attitude determination, these measurements must be brought up to date.

The satellite's body rate, ω, is estimated at every timestep of the GNC system Δt_{GNC} and stored in a buffer of fixed size. When a new star tracker solution q_{star} is available, it is brought up to date using simple Euler integration:

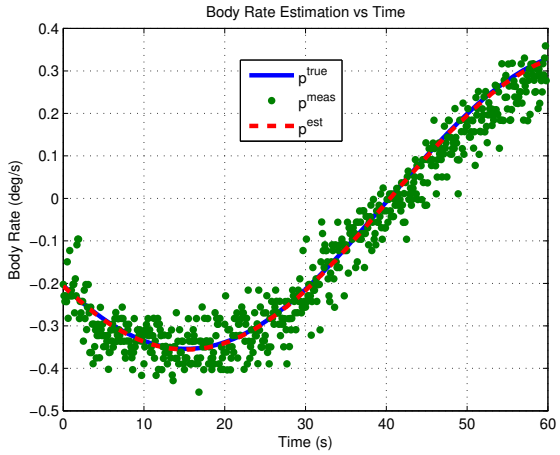

Figure 2. Body Rate Estimation.

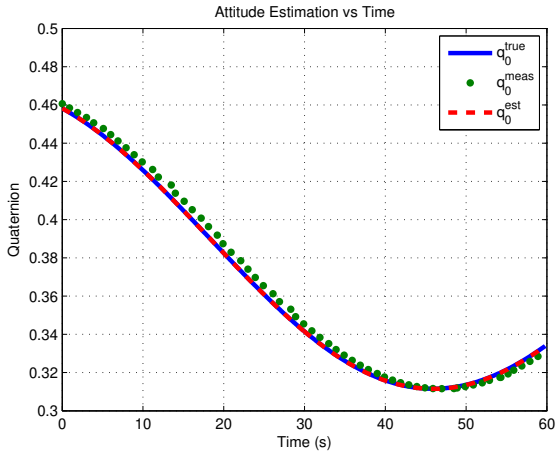

Figure 3. Attitude Estimation.

$q := q_{star}$
$t := t_{star}$
while $t < t_{now}$ **do**

$$\dot{q} := \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega(t) \end{bmatrix}$$

$q := q + \dot{q}\Delta t_{GNC}$
$t := t + \Delta t_{GNC}$
end while

Propagating Star Tracker Solutions

Because several GNC timesteps may go by without a solution from the star tracker, it is necessary to propagate the last solution (after being brought up to date) forward in time. This is done using the current body rate estimate, ω, to Euler integrate forward by one GNC timestep:

$$\dot{q} := \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega \end{bmatrix}$$

$$q := q + \dot{q}\Delta t_{GNC}$$

Figure 3 shows the first quaternion element estimate, truth, and the measured star tracker data over a one minute interval.

4. Orbit Propagation

The Orbit Propagator used within GNC is based on a lower order and degree version of the gravity model used in the Truth Model described in Section 2.

The gravity model returns the gravity acceleration vector g_i in the rotating earth-fixed frame as a function of position in the Earth-fixed frame. Using this model, it is possible to bring GPS measurements up-to-date as well as propagate them forward in time using the following algorithm:

$v := v_{old}$
$x := x_{old}$
$t := t_{old}$
while $t < t_{now}$ **do**
$g_i := g_{model}(x)$
$g := g_i - 2\omega_e \times v - \omega_e \times (\omega_e \times x)$
$x := x + v\Delta t_{orbit} + \frac{1}{2}g\Delta t_{orbit}^2$
$v := v + g\Delta t_{orbit}$
$t := t + \min(\Delta t_{orbit}, t_{now} - t)$
end while

ω_e is the angular velocity of the Earth (typically $\sim 7.29 \times 10^{-5}$ rad/s around the z axis) and Δt_{orbit} cannot be arbitrarily small given the computational constraints placed on the GNC system. If GNC has received a new GPS measurement, then v_{old} and x_{old} will be velocity and position measurements and t_{old} will be the associated timestamp. If there is no new GPS measurement, then all of those values simply carry over from the previous GNC solution and are propagated forward in time by one GNC timestep.

5. Control

Linear quadratic regulators (LQR) are widely used as optimal feedback controllers because of their guaranteed stability properties and desirable performance characteristics. Specifically, for a linear SISO system in which perturbations to gain and phase lag are injected into the state feedback, the gain margins for an undamped system will be better than -6 dB and the phase margin will be at least $\pm 60°$.[7]

The LQR used on SCOUT is based on a linear approximation of the satellite's dynamics while perfectly tracking an attitude command.

The following rotations are defined:

- R_S^I: Rotation from satellite's body axis to inertial
- R_C^S: Rotation from command to the satellite's body axis
- R_I^C: Rotation from inertial to the desired body axes
- θ_x^y: Rotation from x to y frame in axis-angle notation

The rotation from commanded attitude to the satellite's body axis is:

$$R_C^S = [R_S^I]^T R_C^I$$

Taking the time derivative:

$$\dot{R}_C^S = [\dot{R}_S^I]^T R_C^I + R_I^S \dot{R}_C^I$$

A rotation matrix's time derivative can be related to an angular velocity tensor. W_B^{BA} is defined to be the angular

velocity of the B frame relative to the A frame measured in B.

$$\dot{R}_B^A = R_B^A W_B^{BA}$$

Now we can look at the rotation rates in terms of more familiar angular velocity tensors:

$$R_C^S W_C^{CS} = [R_S^I W_S^{SI}]^T R_C^I + R_I^S R_C^I W_C^{CI}$$

$$R_C^S W_C^{CS} = -W_S^{SI} R_I^S R_C^I + R_I^S R_C^I W_C^{CI}$$

And noticing that while tracking well $R_C^S \approx R_S^C \approx I$

$$W_C^{CS} \approx -W_S^{SI} + W_C^{CI}$$

This allows us to relate the angular accelerations of the commanded body frame relative to the body frame, the commanded body frame relative to the inertial frame, and the body frame relative to the inertial frame (where our control effectors create angular accelerations):

$$\dot{\omega}_C^{CS} = \dot{\omega}_C^{CI} - \dot{\omega}_S^{SI}$$

ω_S^{SI} is an the satellite's body rate which is estimated using the methods in Section 3.

The goal of the feedback controller is to stabilize the system in such a way that $R_S^C = I$, since this is the condition where the command is being tracked. At this condition, $\omega_C^{CS} \approx -\dot{\theta}_C^S$ is a reasonable approximation [6] that allows the following simple relationship between the rotation vector derivative and body-angular velocity to be derived:

$$\dot{\theta}_C^S \approx -\omega_C^{CS}$$

$$\dot{\theta}_C^S \approx \omega_S^{SI} - \omega_C^{CI}$$

This approximation is used to build up the simple design model:

$$x = \begin{bmatrix} \theta_{track} \\ \theta_C^S \\ \dot{\theta}_C^S \end{bmatrix}$$

The fictitious state θ_{track} is designed so that, when stable ($\dot{\theta}_{track} = 0$), the system behaves like a first order exponential decay:

$$\dot{\theta}_{track} = \tau\theta_C^S + \dot{\theta}_C^S = 0$$

This gives the following:

$$\dot{x} = \begin{bmatrix} \tau\theta_C^S + \dot{\theta}_C^S \\ \dot{\theta}_C^S \\ \ddot{\theta}_C^S \end{bmatrix}$$

$$A = \begin{bmatrix} 0_{3\times3} & \tau I_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}$$

$$B = \begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ I_{3\times3} \end{bmatrix}$$

Q and R matrices are chosen to inform the LQR solution how to emphasize tracking of states and penalize control effort.

Q is chosen to emphasize stabilization of θ_{track}

$$Q = \begin{bmatrix} \sigma I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}$$

where σ is tuned to give acceptable performance.

The R matrix is chosen to make the control effort penalty commensurate with the expected wheel speed command needed to create the desired acceleration:

$$S = I_{wheel}^{-1} I_{sat}$$

$$R = S^T S$$

Now the algebraic Riccati equation can be solved for our control gains, K_{lqr}:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

$$K_{lqr} = R^{-1}B^T P$$

This controller creates $\dot{\omega}_S^{SC}$ commands that should optimally stabilize θ_C^S near $\theta_C^S = 0$. This must be converted into a wheel acceleration command since the satellite's acceleration cannot be directly commanded:

$$\dot{\omega}_S^{SI} = -K_{lqr} \begin{bmatrix} \int (\tau\theta_C^S + \dot{\theta}_C^S)\delta t \\ \theta_C^S \\ \dot{\theta}_C^S \end{bmatrix}$$

The relationship between $\dot{\omega}_{wheel}$ and $\dot{\omega}_{sat}$ is:

$$\begin{aligned} 0 =& I_{sat}\dot{\omega}_{sat} + \omega_{sat} \times I_{sat}\omega_{sat} \\ &+ I_{wheel}\dot{\omega}_{wheel} + \omega_{sat} \times I_{wheel}\omega_{wheel} \end{aligned}$$

Using the above, it is possible to solve for $\dot{\omega}_{wheel}$:

$$\begin{aligned} \dot{\omega}_{wheel} = -I_{wheel}^{-1} \Bigg[& I_{sat} \Bigg(-K_{lqr} \begin{bmatrix} \int (\tau\theta_C^S + \dot{\theta}_C^S)\delta t \\ \theta_C^S \\ \dot{\theta}_C^S \end{bmatrix} \\ & + \dot{\omega}_C^{CI} \Bigg) + \omega_{sat} \times \left(I_{sat}\omega_{sat} + I_{wheel}\omega_{wheel} \right) \Bigg] \end{aligned}$$

The acceleration of the commanded body frame relative to the inertial frame, $\dot{\omega}_C^{CI}$, can be thought of as a feedforward command generated by the targeting system and fed into the controller. If the target is stationary in inertial space (e.g., a star), then the feedforward command is zero.

Within SCOUT's GNC system, attitude is parameterized as a quaternion while the feedback term in the controller itself is required to be in axis-angle notation. The conversion to an axis-angle vector from a quaternion is done at every GNC timestep and can be found in [6] among other places.

Local stability of the closed loop system is only guaranteed while the satellite is near its commanded attitude. Global stability is suggested through simulation. Figure 4 shows the the angular control error for 100 simulation runs. For these runs, initial attitude was varied with initial yaw, pitch, and roll being uniform random variables between $\pm 180^\circ$ and initial angular velocity varied with initial ω_x, ω_y, and ω_z being uniform random variables between $\pm 50\frac{^\circ}{s}$.

Figure 5 shows a more realistic step response scenario. This time, the simulation is run with no initial body rate (e.g., commanding a new attitude after the satellite has already settled out). As expected, the attitude error looks similar to a first order exponential delay.

For simplicity, the simulations for Figures 4 and 5 are run without the effects of controller discretization and effector saturation.

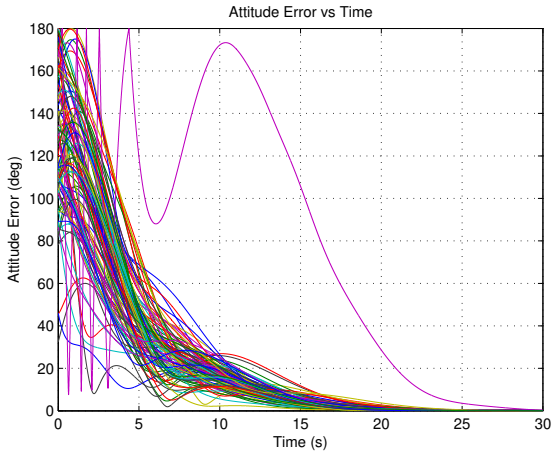

Figure 4. Closed Loop Step Response with Large Initial Body Rate.

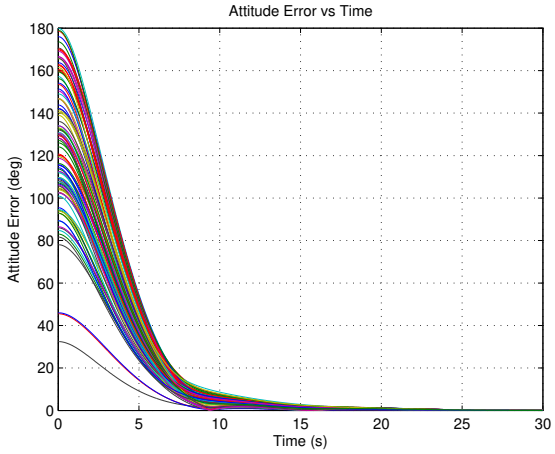

Figure 5. Closed Loop Step Response with No Initial Body Rate.

6. WHEEL DE-SPIN LOGIC

Three orthogonal torque rods on the SCOUT spacecraft are used to dump momentum into the Earth's magnetic field so that the reaction wheels can be spun down without a change in spacecraft attitude.

Powering the torque rods severely interferes with SCOUT's 3-axis magnetometer. To account for this issue, magnetometer readings and torque rod activity are staggered. The torque rods are powered for 27 seconds, then shut completely off for 3 seconds, and then the cycle repeats. During the 3-second window, the magnetometer is de-gaussed and measurements stream into the Flight Computer. Only the final measurement within that 3-second window is actually used. During the 27 second window, the previously measured B field is corrected for using estimates of body angular rate. Differences due to the translation of the satellite within Earth's B field are relatively small and are not accounted for. There is no need for an on-board magnetic field model.

Body rate estimates are used to integrate the quaternion that converts between the body coordinates at the time of the most recent magnetometer measurement and the current body coordinates. This algorithm is shown below:

if (new magnetometer reading) **then**
$q := [1 \quad 0 \quad 0 \quad 0]^T$
$B_{b_0} := B_{sensor}$
else

$$\dot{q} := \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega(t) \end{bmatrix}$$

$q := q + \dot{q}\Delta t_{GNC}$
$B := R(q)B_{b_0}$
end if

Assuming the primary attitude control is able to cancel torque disturbances (τ), the following will typically be true:

$$sign(\tau) = sign(\dot{\omega}_{wheel})$$

For the reaction wheels to despin, the following must be satisfied:

$$sign(\dot{\omega}_{wheel}) = -sign(\omega_{wheel})$$

A simple control law to ensure this is always the case is:

$$\tau = -\hat{\omega}_{wheel} = -\frac{\omega_{wheel}}{|\omega_{wheel}|}$$

The torque produced by the torque rods is

$$\tau = \mu \times B$$

Where μ is the dipole moment of the torque rod. Using the simple control law:

$$-\hat{\omega}_{wheel} = \mu \times B$$

In actuality, it is only possible to oppose the component of $-\hat{\omega}_{wheel}$ which is normal to B. This component is:

$$-\hat{\omega}^{\perp}_{wheel} = -\hat{\omega}_{wheel} + (\hat{\omega}_{wheel} \cdot \hat{B})\hat{B} = \mu \times B$$

Now the vector triple product can be used to solve for μ:

$$B \times (\mu \times B) = B \times [-\hat{\omega}_{wheel} + (\hat{\omega}_{wheel} \cdot \hat{B})\hat{B}] = \mu(B \cdot B) - B(B \cdot \mu)$$

Enforcing μ and B to be orthogonal:

$$B \times (\mu \times B) = B \times [-\hat{\omega}_{wheel} + (\hat{\omega}_{wheel} \cdot \hat{B})\hat{B}] = \mu |B|^2$$

Knowing that, the B vector is normalized:

$$\mu = \hat{\omega}_{wheel} \times B$$

From here, μ can be converted to a command to each of the three torque rods using application-specific scaling and conversion.

SCOUT's torque rods are not powered in modes which require precise pointing because of the large disturbance torques that can be created. The torques induced by the magnetic torque rods are not factored into the attitude control logic mentioned earlier in this paper. Also, once the reaction wheel speeds are below a predefined threshold, the de-spin logic is disabled.

7. PROCESSOR IN THE LOOP TESTING

The GNC system was designed using discrete-time transfer functions, state-space models, and integrators with an eye towards eventual autocoding and running on the flight hardware platform.

The equations of motion, effector model, and sensor models are run on a desktop Simulink® platform. Sensor outputs are pushed out at 20 Hz over a UDP interface.

GNC is autocoded into C, compiled, and then executed on the flight hardware platform. The UDP interface is checked periodically for new sensor data and fed into GNC, which iterates and provides new effector commands. These commands are sent back to the desktop simulation via UDP. The process is summarized in Figure 6.

This setup allows us to look into the execution time of the GNC system. Figure 7 shows the execution time for 1000 iterations of GNC. The mean execution time is ~20 ms which is well below GNC's designed update period of 100 ms.

Lessons Learned

Processor in the loop testing led to a number of improvements and simplifications to the GNC system which shortened total execution time.

The Earth-Centered-Inertial to Earth-Centered-Earth-Fixed transformation was changed from a complex calculation run at every timestep to a quaternion-based lookup table that is calculated a-priori. The table is designed to be valid for the entire expected orbital lifetime of the satellite.

The onboard Kalman filter was simplified to be linear and constant-gain to avoid computationally-costly matrix inversions. Instead of estimating attitude within the Kalman filter the star-tracker provided attitude estimates are brought up-to-date using the angular rate estimates.

Finally, sensor inputs are sanitized to be within reasonable ranges to protect against the potentially unforeseen consequences of "garbage" data being unintentionally passed into GNC. Rarely, GNC executation times of > 1 sec were noticed and determined to result from very slow execution of the `pow` function within `libc` when nonsense data was passed to GNC. While unexpected in normal operation, a floating A/D signal or broken GPS receiver could lead to this type of behavior. If data is determined to be out-of-bounds it is flagged in SCOUT's health log so that the root cause can be diagnosed.

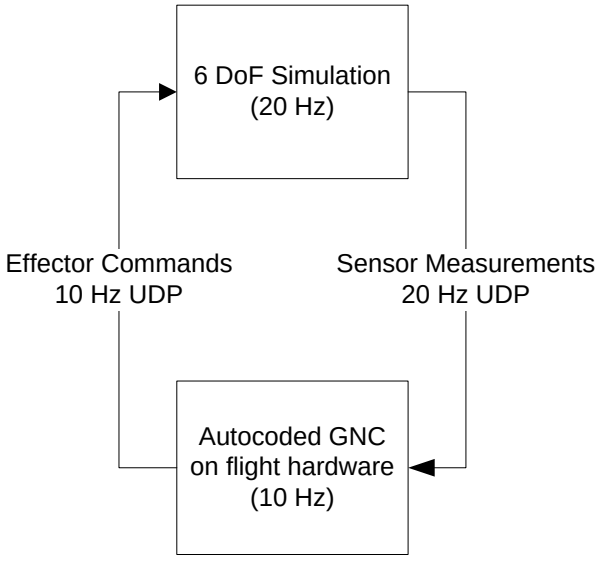

Figure 6. Processor in the Loop Flow.

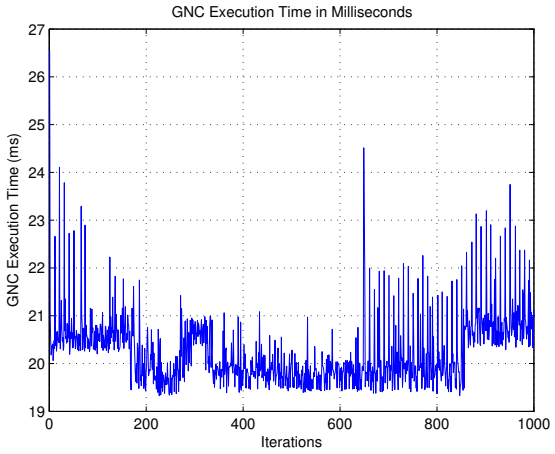

Figure 7. GNC Execution Time.

8. SUMMARY

This paper provides an overview of the simulation environment, GNC solution, and processor-in-the-loop testing of the SCOUT spacecraft being designed by Andrews Space. Confidence in the 6-DoF simulation was gained by its use on several prior projects without issue. The 6-DoF simulation uses established equations of motion and, where possible, effector models are proven against experimental data.

Confidence in the GNC solution is gained through closed-loop simulation as well as designed-in stability margins. A linear constant-gain Kalman filter was developed to provide adequate state estimation while avoiding the computational complexities of Kalman filters which require matrix inversion. For orbit propagation, a low order gravity model is included and numerically integrated to bring GPS solutions up to date. The controller is a linear quadratic regulator whose global stability is suggested through simulation.

Even with the GNC system running acceptably in simulation, it was important to make sure the execution time on the embedded flight hardware is not overly taxing on system resources. It was shown that GNC's execution time was considerably smaller than the designed discretization timestep of 0.1 seconds.

Ultimate verification of the GNC solution and simulation will be demonstrated after SCOUT's launch in mid-2015. The 6-DoF model will be compared against data measured on-orbit and, if needed, updates to SCOUT's control and estimator gains will be uploaded to the spacecraft.

Several improvements to SCOUT's GNC system are currently being investigated. Among them, implementation of an LQR-based control which is valid for saturated actuators (SLQR),[8] addition of a multiplicative EKF for attitude estimation (possible with improvements to the flight computer

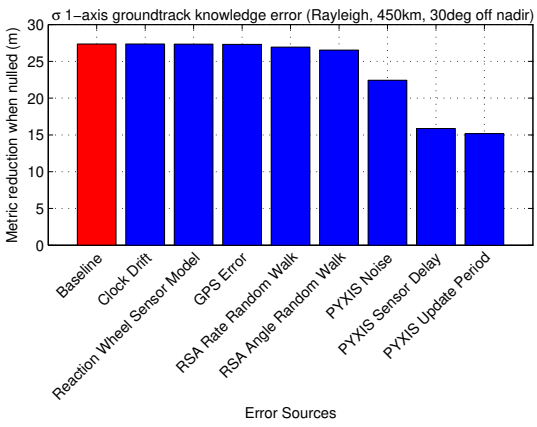

Figure 8. Knowledge Error Sensitivity.

processing capability), and on-board estimation of sensor misalignments (currently done on the ground).

Sensitivity of SCOUT's groundtrack knowledge accuracy to various error sources was investigated using the simulation. Eight different error sources were included in the study:

- Clock drift from true time
- Variable time delay in the reaction wheel speed sensor
- GPS noise
- Gyro rate random walk
- Gyro angle random walk
- Star tracker (PYXIS) noise
- Star tracker (PYXIS) sensor delay
- Star tracker (PYXIS) update period

The results are summarized in Figure 8. Through simulation we expect the baseline groundtrack knowledge error to have a 1-σ 1-axis value of $\sim$27 m at an altitude of 450km while tracking a target 30° off-nadir (neglecting the effect of sensor/camera misalignments). Each blue bar shows the effect of removing one error source. The results indicate that star tracker (PYXIS) error sources have the most dominant effect on knowledge accuracy. Consequently, star tracker throughput and delay have been flagged as desirable future improvements.

REFERENCES

[1] G. H. Kaplan, "The IAU resolution on astronomical constants, timescales, and the fundmental reference frame," *US Naval Observatory Circulars*, vol. 163, 1981.

[2] C. Finlay, S. Maus, C. Beggan, T. Bondar, A. Chambodut, T. Chernova, A. Chulliat, V. Golovkov, B. Hamilton, M. Hamoudi *et al.*, "International geomagnetic reference field: the eleventh generation," *Geophysical Journal International*, vol. 183, no. 3, pp. 1216–1230, 2010.

[3] M. Sidi, *Spacecraft Dynamics and Control: A Practical Engineering Approach*, ser. Cambridge Aerospace Series. Cambridge University Press, 1997. [Online]. Available: http://books.google.com/books?id=xQpZJMtDehQC

[4] R. G. Gottlieb, *Fast gravity, gravity partials, normalized gravity, gravity gradient torque and magnetic field: derivation, code and data*, 1993, vol. 1.

[5] C. S. C. A. S. Operation and J. Wertz, *Spacecraft Attitude Determination and Control*, ser. Astrophysics and Space Science Library : a series of books on the recent developments of space science and of general geophysics and astrophysics. Reidel, 1978. [Online]. Available: http://books.google.com/books?id=GtzzpUN8VEoC

[6] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," 2006.

[7] C. Zhang and M. Fu, "A revisit to the gain and phase margins of linear quadratic regulators," *Automatic Control, IEEE Transactions on*, vol. 41, no. 10, pp. 1527–1530, Oct 1996.

[8] C. Gkcek, P. T. Kabamba, and S. M. Meerkov, "Slqr/slqg: An lqr/lqg theory for systems with saturating actuators," *CGR-00-03, EECS, UNIVERSITY OFMICHIGAN, ANN ARBOR*, vol. 46, pp. 1529–1542, 2000.