Simulation of the PolarCube Attitude Control and Determination Subsystem

Franklin Hinckley
Advisor: Brian Sanders
University of Colorado - Boulder
franklin.hinckley@colorado.edu
April 23rd, 2016

Abstract

The PolarCube mission currently under development by the Colorado Space Grant Consortium at University of Colorado - Boulder requires a relatively complex and high-performance Attitude Determination and Control Subsystem to meet the requirements of the mission. As accurately testing an integrated ADCS on Earth is quite difficult, since the testing fixture interferes with the system response, the verification of the system is principally reliant on simulation, at least for assessing performance. To address this a high-fidelity simulation of the subsystem was developed and tested. This simulation shows that the ADCS design, subject to realistic actuator constraints and sensor performance, does meet the requirements of the mission.

1 Background

The PolarCube mission from Colorado Space Grant Consortium, in collaboration with the Center for Environmental Technology and the National Snow and Ice Data Center, is a 3U CubeSat housing a 118GHz radiometer for atmospheric temperature profile measurement. This instrument requires a rotating parabolic mirror to provide the horizontal scanning that builds up the data collection swath as it sweeps through pointing nadir and for calibrating against the temperature of space when pointing zenith. This mirror, as the effect of its rotation is to make the full system inertia variable, creates several potential concerns for the ADCS subsystem. As the pointing error and knowledge error requirements for science collection are fairly tight, 2.3° and 0.23° respectively, the possible ADCS problems from the mirror are very difficult to assess outside of simulation. A physical test setup would need to provide measurements to this precision and the interference of the test setup with the natural dynamics would also need to be less than the requirements. This combination of relatively tight, for a CubeSat, pointing requirements and a difficult control application makes for a complex ADCS design and one that is difficult to assess outside of simulation. This requires that the simulation accurately model the effects of the mirror and include sufficient fidelity models of other components to make the simulation results usable.

Figure 1: PolarCube Satellite
2 Satellite ADCS Overview

The Attitude Determination and Control Subsystem for PolarCube consists of three major elements: the driver cards, main board, and star tracker. Each of the three driver “cards” in the reaction wheel assembly mounts a reaction wheel, a magnetic torque rod, and the driver board PCB that translates the outputs of the main control board into the inputs to the actuators. The structural cards, wheels, torque rods, and PCBs are custom-designed by Colorado Space Grant Consortium. The motors used for the ADCS reaction wheels are the Faulhaber 0620K012B three-phase brushless DC motor [1]. The main board houses the two fully-redundant MEMS gyroscopes and two fully-redundant MEMS magnetometers along with the main subsystem processor. The star tracker board, at the right of the assembly shown below, houses the image sensor and dedicated star tracker processor along with the required support circuitry for the processor. Both the main board and star tracker are custom-designed by Space Grant.

![Figure 2: PolarCube ADCS](image)

3 Attitude Representation

Before describing the control law and attitude determination filter the selected attitude representation should be discussed. The Modified Rodrigues Parameters (MRPs) were selected as they are a minimal representation, the singularity can be easily avoided, and they are commonly used in the graduate program at CU-Boulder so incoming students are familiar with their use.

The MRPs can also be computed from the quaternions $\beta$ using Equation (2), where $\beta_0$ is the scalar component of the quaternion.

$$\sigma_i = \frac{\beta_i}{1 + \beta_0}$$  (2)
It is important to note here that the two valid quaternion representations, the long and short rotations which differ by a sign, of a given attitude will give different MRPs. This introduces the concept of the "shadow set" of MRPs. Using the other quaternion gives:

\[ s \sigma_i = \frac{-\beta_i}{1 - \beta_0} \] (3)

As with any minimal attitude representation the MRPs will have singularities, however the existence of a distinct shadow set allows the singularity to be avoided with minimal effort. The primary set of MRPs have a singularity when \( \beta_0 = -1 \), i.e. 360°, and the shadow set have a singularity when \( \beta_0 = 1 \), i.e. 0°. This means that by switching back and forth between the sets, i.e. redefining the current rotation angle, the singularity can be avoided. The mapping between the two sets is:

\[ s \sigma = \frac{-\sigma}{\sigma^T \sigma} \]

The typical switching surface is when the norm of the MRP reaches one, which occurs at a rotation of 180°. At this point the norm of the primary set is increasing past one towards the singularity while the norm of the shadow set is decreasing past one from its singularity. When the norm of the shadow set reaches one the same mapping is used to go back to the original set, as it is how the shadow set for the current attitude measure. This switching condition also ensures that the attitude measure is never closer than 180° to a singularity and makes selection of control gains easier as the magnitude of the attitude representation is bounded to one. It is important to note that switching at a norm slightly greater than one is not a problem, as the singularity is not encountered until 180° past the switching point, so there is no need for a more complicated check inside the integrator to ensure the norm does not exceed one.

3.2 Kinematics

The kinematic differential equation for the MRPs is given by:

\[ \dot{\sigma} = \frac{1}{4} \left[ \left( 1 - \sigma^T \right) [I_{3 \times 3}] + 2 \left[ \sigma \right] + 2 \sigma \sigma^T \right] \omega = \frac{1}{4} [B(\sigma)] \omega \] (4)

Assigning explicit frames for the case where the MRP \( \sigma \) represents the attitude of the spacecraft body frame relative to inertial space:

\[ \dot{\sigma} = \frac{1}{4} \left[ \left( 1 - \sigma^2 \right) [I_{3 \times 3}] + 2 \left[ \sigma \right] + 2 \sigma \sigma^T \right] B \omega_{B/N} = \frac{1}{4} [B(\sigma)] B \omega_{B/N} \] (5)

The kinematic differential equation is the same for both the primary and shadow sets.

3.3 Relative MRPs

One of the terms that will come up in the control is the attitude of the satellite relative to the reference. The MRP for the body relative to inertial space and the MRP for the reference relative to inertial space are not from consistent frames so cannot be simply subtracted to obtain the MRP of the body relative to the reference. There is an equation for directly computing the relative MRP but as there are singularities in the MRPs care must be taken when operating on the MRPs. A more robust algorithm, that is guaranteed to avoid the singularities and always provide the MRP for the short rotation, is summarized below.

1. Convert both MRPs to the corresponding Direction Cosine Matrix (DCM)
2. Compute the DCM for the relative orientation of the body frame to the reference
3. Convert the relative DCM to quaternions using Sheppard’s Method, the implementation of Sheppard’s Method should return the short rotation
4. Convert the quaternion to an MRP

This method is computationally expensive but is the most robust method of computing the MRP for the attitude of the body relative to the reference. A more integrated implementation, instead of the discrete subroutines currently used in the simulation, is being developed and may allow for improved computational efficiency while avoiding the potential singularities of the direct method.
4 Dynamics

Before introducing the complexities of a controlled system or discussing the implementation of the ADCS simulation the first step is to derive the system dynamics.

The first definitions required are the frames necessary to define the problem and the resulting inertias in those frames. These frames are the primary satellite frame, the frames of the reaction wheels, and the frame of the mirror.

As there is no mirror principal axis aligned with the rotation axis and there is not an obvious axis of symmetry the mirror frame is selected to have its z-axis aligned with the rotation axis and its first axis aligned with the edge of the mounting boss. The frame is then defined by \( \mathcal{M} : \{ \hat{\mathbf{m}}_s, \hat{\mathbf{m}}_l, \hat{\mathbf{m}}_r \} \) where \( \hat{\mathbf{m}}_s \) is qualitatively the short transverse axis of the mirror, \( \hat{\mathbf{m}}_l \) is qualitatively the long transverse axis of the mirror, and \( \hat{\mathbf{m}}_r \) is the rotation axis. As the mirror is not square to its mount the transverse axes of \( \mathcal{M} \) are not exactly the long and short axes but this does suffice to identify which is which. The trade-off for the selection of the mirror frame was between a diagonal inertia tensor with an odd rotation between the mirror principal frame and a mirror frame with an axes aligned with the rotation geometry and the decision was made to accept a fully-populated inertia tensor in exchange for a mirror frame aligned with the rotation axis. The mirror frame is illustrated in Figure (3).

![Figure 3: Mirror Frame](image)

The satellite frame has its first axis aligned with the transmit antenna, the second axis aligned with the star camera lens, and the third axis aligned with the payload to define the body frame as \( \mathcal{B} : \{ \hat{\mathbf{b}}_t, \hat{\mathbf{b}}_s, \hat{\mathbf{b}}_p \} \). As the satellite mass distribution is also not uniform about the geometric axes the satellite inertia tensor will also be fully populated. As all operations are defined by the geometric axes, e.g. point the antenna at the ground station, it makes sense to use the geometric axes instead of the principal axes as the satellite frame for this reason as well as simplifying the rotations between the mirror and wheel frames and the body frame. This frame is illustrated in Figure (4).

![Figure 4: Satellite Body Frame](image)

The final frames are those of the reaction wheels. Each wheel frame is defined as \( \mathcal{W} : \{ \hat{\mathbf{w}}_s, \hat{\mathbf{w}}_{t_1}, \hat{\mathbf{w}}_{t_2} \} \) where \( \hat{\mathbf{w}}_s \) is the spin axis of the wheel and \( \hat{\mathbf{w}}_{t_1} \) and \( \hat{\mathbf{w}}_{t_2} \) are the transverse axes of the wheel, which are arbitrary in the
plane normal to the spin axis due to the rotational symmetry of the wheel so can be considered to be fixed in the body frame. The inertia tensor of the wheels is diagonal as the axes of symmetry selected as the frame axes are the principal axes, the inertia further simplifies as the two transverse inertias are the same. The inertia tensor for a wheel is defined as:

$$W[I_W] = \begin{bmatrix} J_I & 0 & 0 \\ 0 & J_I & 0 \\ 0 & 0 & J_s \end{bmatrix}$$

The final set of definitions required are the angles and angular rates between frames. For the mirror frame both the angle and the angular rate are needed as the angular position of the mirror determines the full spacecraft inertia and the angular rate affects the dynamics. The angle between the intersection of the mirror long transverse axis with the mirror deployment strut is defined as $\mu$, the rotation rate of the mirror is then $\omega_{M/B} = \dot{\mu}\hat{m}_r$. For the rotation of the reaction wheels relative to the body and the rotation of the spacecraft body relative to inertial space only the angular rate is needed. The angular rate of wheel $i$ relative to the body is given by $(\omega_{W/B})_i = \Omega_i\hat{w}_s_i$. The angular rate of the spacecraft body relative to inertial space is simply defined as $\omega_{B/I'}$.

Now that all definitions are known the Equations of Motion can be determined. The final Equations of Motion for the satellite, derived in Appendix A, are:

$$[I] \dot{\omega} = -[\bar{\omega}][I] \omega - \sum_{i=1}^{3} J_{s_i} \left( \Omega_{i}\hat{w}_s_i + [\bar{\omega}] \Omega_i \hat{w}_s \right) + L$$

$$- [\bar{\omega}_{M/B}] \left( [B M]^M[I_M] [B M]^T \right) (\omega + \dot{\mu}\hat{m}_r)$$

$$- [\bar{\omega}] \left( [B M]^M[I_M] [B M]^T \right) (\dot{\mu}\hat{m}_r) - \left( [B M]^M[I_M] [B M]^T \right) (\dot{\mu}\hat{m}_r + [\bar{\omega}] \dot{\mu}\hat{m}_r)$$

While this equation provides the evolution of the satellite angular velocity the evolution of the attitude representation and the wheel speeds are also required. These equations are coupled as the satellite angular velocity and wheel accelerations are dependent on each other. The mirror angular acceleration and rotation rate also appear in the equations but this project assumes that there is a servo loop internal to the payload that enforces the desired mirror rate history and this servo is assumed to be perfect. This assumption allows the simulation to simply use the specified mirror acceleration and rotation rate at each time step. A possible way to evaluate the coupled Equations of Motion is to evaluate the state-space form where the derivative functions can be expressed as the system mass matrix multiplied by the state vector rates. For this spacecraft this takes the following form:

$$\begin{bmatrix} I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{1x3} & I_s & J_s \end{bmatrix} \begin{bmatrix} \dot{\sigma} \\ \dot{\omega} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_\sigma \\ \mathbf{f}_\omega \\ \mathbf{f}_\Omega \end{bmatrix}$$

Where the functions on the right-hand side are given by the following expressions:

$$\mathbf{f}_\sigma = \frac{1}{4} [B (\sigma)]^B \omega$$

$$\mathbf{f}_\omega = -[\bar{\omega}][I] \omega - \sum_{i=1}^{3} J_{s_i} [\bar{\omega}] \Omega_i \hat{w}_s_i + L - [\bar{\omega}_{M/B}] \left( [B M]^M[I_M] [B M]^T \right) (\omega + \dot{\mu}\hat{m}_r)$$

$$- [\bar{\omega}] \left( [B M]^M[I_M] [B M]^T \right) (\dot{\mu}\hat{m}_r) - \left( [B M]^M[I_M] [B M]^T \right) (\dot{\mu}\hat{m}_r + [\bar{\omega}] \dot{\mu}\hat{m}_r)$$

$$\mathbf{f}_\Omega = \mathbf{u}_s$$

This form of the Equations of Motion are easily solved in the simulation by premultiplying both sides by the inverse of the system mass matrix. As the evaluation of the dynamics is only required in the simulation, not the flight code, the fact that evaluating the dynamics is computationally expensive is not a concern.

5 Control Law

The control law used for the PolarCube mission is a nonlinear Lyapunov control developed by Hogan and Schaub [2]. This control is globally asymptotically stabilizing.
5.1 Lyapunov Control

Lyapunov control theory is based on the analysis of the Lyapunov function. This function is a positive-definite scalar function of the system states with continuous partial derivatives. If the derivative of the Lyapunov function is negative semi-definite in all states of interest the system is stable, if the derivative is negative definite in all states of interest the system is asymptotically stable, i.e. will converge to the reference. It is important to note here that this analysis provides no performance information, just that the system is stable. There is no information on the timescale of the convergence. It is also important to note that the derivative of the Lyapunov function not being negative (semi-)definite does not mean that the system is not stable or asymptotically stable, just that the analysis has not shown that it is.

5.2 Lyapunov Functions

The Lyapunov function for the baseline control, used during all modes except science, is defined as:

\[ V(\delta \omega, \sigma) = \frac{1}{2} \delta \omega^T [I] \delta \omega + 2 \ln (1 + \sigma^T \sigma) \]  

In science mode an integral term is added to improve disturbance rejection to help meet the tighter pointing requirements of this mode. The modified Lyapunov function to include this integral term is defined as:

\[ V(\delta \omega, \sigma, z) = \frac{1}{2} \delta \omega^T [I] \delta \omega + 2 \ln (1 + \sigma^T \sigma) + \frac{1}{2} z^T [K] z; \quad z = \int_0^t (K \sigma + [I] \delta \omega) dt \]  

In both equations \( \delta \omega \) is the angular rate error between the true body angular rate and the reference rate and \( \sigma \) is the MRP describing the attitude of the body relative to the reference. The integral term \( z \) is usually implemented as \( z = K \int_0^t \sigma dt + [I] (\delta \omega - \delta \omega_0) \) for computational ease where \( \delta \omega_0 \) is the rate error when the control switches to the integral controller.

5.3 Control Law

When the derivative of the Lyapunov Function is computed the term \([I] \dot{\omega} \) will appear, this can be substituted out using the dynamics. This substitution introduces the control torque, the form of the control torque can then be selected to force the Lyapunov rate to be negative definite. As the control objective is to drive the attitude and rate error to zero the Lyapunov rate should be negative definite instead of just the negative semi-definite required for stability. The final form of the control law is detailed in Equation (11). This form of the control law avoids a quadratic angular velocity term to reduce the commanded control during detumble and maneuvers while retaining asymptotic stability.

\[ L_r = -[I] (\dot{\omega}_r - [\tilde{\omega}] \omega_r) + K \sigma + [P] \delta \omega + [P] [K_r] z - \left( [\tilde{\omega}_r] - [\tilde{K}_r] z \right) (\dot{[I]} \omega + [G_s] h_s) + L \]  

As the reaction wheel spin axis matrix \([G_s]\) for PolarCube is invertible the individual reaction wheel commands are then computed from the system-level required torque using Equation (12).

\[ u_{sl} = [G_s]^{-1} L_r \]  

5.4 Momentum Management

An additional consideration for the control is that the reaction wheels are only capable of finite speed. While maneuvers do not change the wheel speeds, as all momentum transferred to the wheels to execute the maneuver is transferred back to the satellite at the end of the maneuver, the initial detumble of the satellite and compensation for external disturbances result in a net change in the wheel speed. To avoid reaching the maximum speed of the reaction wheels the torque rods are used for momentum management. Conceptually, this process consists of creating an external torque on the satellite using the torque rods and then correcting for this torque using the reaction wheels. The external torque is generated such that the correction torque from the reaction wheels reduces the wheel speeds. This correction is computed using the following, summarized from Hogan and Schaub.
Using a simplification of the equation for the motor torques, \( u_s = [J_s] \dot{\Omega} \), allows the following expression for the wheel speed correction torque.

\[
\mathbf{u}^* = -K_M [J_s] (\mathbf{\Omega} - \mathbf{\Omega}_0)
\]  

As this torque alone will reduce the wheel speeds but impart a net torque on the satellite the torque rods are used to produce a compensating torque. This strategy is complicated by the fact that the torque rods cannot produce torque about the local magnetic field vector so the desired compensating torque may not be possible. As a proper matrix inverse is then not possible an SVD pseudoinverse, denoted by the \( \dagger \) superscript, is used to compute the following solution for the magnetic dipoles.

\[
\mathbf{\mu}^* = -\left( [\tilde{B}] [G_t] \right)^\dagger [G_s] \mathbf{u}^* 
\]

The magnetic torque on the satellite is then:

\[
\tau_M = -[\tilde{B}] [G_t] \mathbf{\mu}^* 
\]

As this torque is not generally the desired compensating torque, due to the aforementioned inability to produce a torque about the local field vector, an additional correction is necessary to make the net torque on the satellite from the momentum management process to be zero. This correction torque is computed using:

\[
\Delta \mathbf{u} = [G_s]^{-1} (\tau_M - [G_s] \mathbf{u}^*) 
\]

The final control output, comprised of the primary control law and the momentum management corrections, is defined in Equation (17).

\[
\mathbf{u}_s = \mathbf{u}_{sl} + \mathbf{u}^* + \Delta \mathbf{u} 
\]

In the simulation this final control torque expression is passed to the dynamics subroutine along with the magnetic torque on the satellite \( \tau_M \) for inclusion with any other external torques in evaluating the dynamics.

### 5.5 Pointing Targets

The final piece for using the control is how the pointing targets are determined. The science pointing target is fully defined, as the mirror beam must be directed nadir and the scan swath must be cross-track, but the S-Band communications and sun-pointing modes do not specify the rotation about the relevant axis. The rotation of the satellite about the antenna boresight or solar panel normal vector does not matter to the primary functionality of those modes so there is a degree of freedom that needs to be determined from another source. The TRIAD method, outlined below, is used to form the full pointing target where \( \hat{t}_i \) are the TRIAD-frame unit vectors and \( v_i \) are the pointing target vectors.

1. \( \hat{t}_1 = v_1 \)
2. \( \hat{t}_2 = \frac{v_1 \times v_2}{|v_1 \times v_2|} \)
3. \( \hat{t}_3 = \hat{t}_1 \times \hat{t}_2 \)

The Direction Cosine Matrix (DCM) for the pointing target relative to inertial space is then formed as \( [C] = [\hat{t}_1 \hat{t}_2 \hat{t}_3] \). The appropriate rotation of the satellite to align the desired satellite axis with this frame is then applied and the resulting DCM is converted to the reference MRP for use in the control.

The other reference considerations in the control are the reference angular velocity and angular acceleration. The reason for including these in the control is that otherwise the control response to drive the attitude to the reference and then stop there. If the pointing target is time-varying then the tracking performance can be improved by driving to an angular rate that is consistent with the known time series of attitude targets. However, computing this reference trajectory can be computationally expensive, particularly for cases like tracking a fixed ground target where the “shape” of the pass varies between passes, so the reference angular rate and acceleration are set to zero in the control. The tracking performance analysis shows that the pointing requirements are still met. As the reference terms are still present in the control they could be evaluated, or at least approximated, without changing the control module.
6 Attitude Determination Filter

The purpose of the attitude determination filter is to take in the measurements from the gyroscope and star tracker and return an estimate of the current angular rate and attitude of the satellite.

The primary attitude determination filter is an Extended Kalman Filter based on the work of O’Keefe and Schaub [3] as well as Karlgaard and Schaub [4].

6.1 Gyroscope Filtering

The MEMS gyroscopes on PolarCube provide relatively noisy output so there is some filtering of the measured rates before the primary filter. This is not strictly necessary, as the primary filter is stable without filtering the gyroscope data, but it improves the control performance. Without filtering the gyroscope data the propagated state is correspondingly noisy so the control responds to what it sees as state errors, increasing the control effort and reducing the tracking performance. As the actual angular rates of the satellite are slowly varying simply averaging the gyroscope output over some time interval greatly reduces the noise in the rate estimate. The length of this averaging time is dependent on the required smoothing and acceptable filter delay, as a longer averaging will produce a smoother result but increases the filter delay. If the filter delay is too long then the tracking performance, and attitude knowledge accuracy, is compromised by using ”old” rate information. The best balance between smoothing and small delay changes depending on the requirements of the specific mode and is discussed further in the configuration section.

6.2 Extended Kalman Filter

The primary attitude determination filter is an Extended Kalman Filter (EKF). The EKF is a sequential filter, i.e. updates the state estimate as new information becomes available so is suitable for real-time use. For this application the states of interest are the current attitude representation and the rate measurement biases of the gyroscopes. As it is not possible to use the information from both gyroscopes in the same filter while estimating the biases of the separate gyroscopes the two gyroscopes are averaged together and treated as a composite sensor. The estimated bias is then the bias of the composite sensor, which is simply the average of the individual biases. As the gyroscope measurement biases are assumed to be constant the estimated state vector is \( \mathbf{x} = [\mathbf{\sigma} \, \mathbf{\omega}_b]^{\top} \). The gyroscopes do have some drift (variation of the bias) over time but this effect is small enough that the update of the state vector will correct for this drift without explicitly estimating the drift in the filter.

The primary steps for a sequential filter are the time update and the measurement update. The time update maps the state estimate and state covariance matrix forward from the time of the previous estimate to the current time. The measurement update then corrects the state estimate at the current time based on some measurement data. The EKF then updates the nonlinear trajectory using this new state estimate. While the EKF does update the nonlinear trajectory as part of the measurement update the filter is still operating on the linearized system dynamics. As the MRPs linearize well this is not a concern but it is still important to recognize that this is what is happening inside the filter. The EKF as applied to attitude determination in MRPs has the interesting property, in contrast to other applications of the EKF such as orbit determination, that it does not need to initialized using a Conventional Kalman Filter. This is due to the fact that the possible states are constrained, as the norm of the MRP is bounded to one, and that neighboring trajectories remain neighboring, due to the linearity of the MRPs, so errors in the reference trajectory do not cause the filter solution to diverge.

When using MRPs as the attitude representation there are some additional considerations due to the set switches. The state estimate and state covariance must be correctly mapped through the set switches and the measurement of the attitude can be represented in either set. The papers by O’Keefe and Schaub [3] and Karlgaard and Schaub [4] provide an excellent discussion of the EKF as applied to the attitude determination problem in MRPs and are the basis for the PolarCube filter. Their work and some of the implementation details are summarized in Appendix B.

The control needs updated state estimates at the same rate that the control is updated, otherwise the new control output will be the same as the previous time and the effective update rate is the slower measurement rate. As the star tracker only provides new attitude solutions every few seconds and the control may be updated at several Hertz the standard filter implementation of only mapping to the current measurement time and updating will not work. However, as the time update does use the current gyroscope measurements the filter can be
implemented with the time update step evaluated at the control update rate and then the measurement step evaluated when a new star tracker solution is available. This does mean that the estimated angular rate is based on the gyroscope bias estimate from the previous update but as the gyroscope bias is constant, or at worst slowly drifting, this is not an issue. As the gyroscopes are fairly noisy the intermediate state estimates will show some variation but this is addressed by the gyroscope filtering.

7 Simulation Structure

Since the design of the Attitude Determination and Control Subsystem will be worked on by various students over the life of the project the simulation needs to be user-friendly as well as accurate. The simulation is built out of functional modules, e.g. the integration of the dynamics is packaged in its own subroutine, so that the major functional steps of the simulation are separate from the low-level details. This allows a new engineer to learn the sequence of the simulation before needing to understand the details of a component. This structure also allows for modules to be easily swapped out for cases like testing a different control implementation and allows for the simulation modules for flight code to be replaced with the actual flight code for software-in-the-loop testing. Another benefit of this structure is that it allows sections of the simulation to be verified and then left alone, allowing new functionality to be easily built on the existing code while avoiding the concerns of new additions compromising the previous work.

The simulation modules are divided into three major functional groups: simulation, ADCS flight software, and system flight software. The simulation modules are pieces like integrating the dynamics that handle computing the stimuli for ADCS and evaluating the system response. The ADCS flight software modules are pieces such as evaluating the control law that will be part of the flight code. The system flight software modules are flight code that will not be running on ADCS; these cover configuration settings like the control gains and effectively replace the ADCS server in the main satellite software. This division makes translating the ADCS simulation into flight code straightforward as functionality is already grouped by where it will go. This also makes code development clearer as it ensures that there is no mixing of functional groups that would require restructuring when writing the flight code.

8 Key Modules

While it is not necessary to discuss every piece of code that is part of the simulation it is valuable to cover the key modules.

Simulation Modules

- simSettings: provides a single configuration point for simulation duration and start time, orbit definition, and plotting control
- ADCS_Sim: main simulation function, handles passing information between subroutines, logging states, and computing verification values
- ADCS_Plot: produces the requested plots
- ADCS_Integrate: evaluates the dynamics to determine the updated attitude, satellite angular rate, wheel speeds, and mirror position
- ST_Model: produces a simulated star tracker solution given the current true attitude and checks the angular rate limit for a successful lock
- A3G4250_Model: produces simulated gyroscope measurements for each gyroscope given the current true angular rate and specified gyroscope parameters

ADCS Modules

- ADCS_Control: implements the control law
- ADCS_Filter: implements the primary attitude determination filter (EKF)
• ADCS_Prefilter: handles the smoothing of the gyroscope output before calling the main filter

• satProps: defines the mass properties and actuator axes, also defines the sensor model specifications used by the simulation modules

FSW Modules

• modeCheck: checks the current satellite position and time to determine the current mode

• modeParamSet: sets the configuration settings, e.g. control gains, for the current mode

9 Sensor Performance and Actuator Limits

To ensure that the performance realized by the simulation is representative of the physical system the assumed quality of the simulated sensor inputs must be representative of the performance of the actual sensors. This is particularly important for assessing the performance and stability of the attitude determination filter. In addition to the sensors listed below there are also the reaction wheel encoders, for wheel speed measurement, and the magnetometer. These are not currently modeled in the simulation so their outputs are assumed to be perfect. As the encoders provide 50 counts per revolution and the control only needs the speed, not angular position, assuming this output to be perfect is a reasonable approximation. The magnetometer output is certainly not perfect but as the magnetic field measurement is only used for momentum management this is not considered to invalidate the simulation results. The specifications for the gyroscopes and star tracker are:

1. Gyroscope Noise: both gyroscope models use a noise standard deviation of \([0.1328, 0.1203, 0.1385]^\top \text{deg/s}, \text{for the three axes, as per a static test log of the gyroscope}\)

2. Gyroscope Bias: the first gyroscope model uses a bias of \([-0.0137; 0.2418; -0.0202]^\top \text{deg/s}, \text{the second uses the same numbers in a different order with different signs. The first bias is from the same test log as the noise statistics and the second bias was set to the same magnitude but different values to test if the bias of the composite gyroscope is correctly estimated by the filter.}\)

3. Star Tracker Noise: the star tracker MRP solution is assumed to have 100 arc-seconds of noise, there is currently no test data for the accuracy of our module so an initial approximation of five times the 20 arc-seconds realized by a COTS part (Blue Canyon’s XACT) is used.

It is also critical in the assessment of the simulation results to ensure that no limits of the physical system are exceeded, else the performance realized in simulation is meaningless. The principal limitations of the system are summarized below:

• Reaction Wheel Maximum Torque: the Faulhaber 0620K012B motors are rated for continuous operation at 0.30 mNm, the command torques are limited to 90% of this (0.27 mNm)

• Reaction Wheel Maximum Speed: the Faulhaber motors are rated for continuous operation at 7290 rpm, the limit is considered to be 90% of this (6561rpm)

• Star Camera Maximum Rate: the star tracker cannot reliably achieve a solution if the angular rate is higher than 0.25°/s

• Control Update Rate: current test data for running the control on the flight hardware shows a maximum control update rate of approximately 1kHz

There will also be a limit on the maximum dipole moment of the torque rods but a torque rod model has not yet been implemented in the simulation. There is also a duty-cycle limit on the torque rods as the de gauss profile must be run before sensing with the magnetometer, otherwise the magnetometer will just read the residual field in the torque rods, but the duty cycle on the torque rods will be around 90%. As the torque rods are only needed for momentum management, not attitude control, a possible overestimate of torque rod performance in the current simulation does not significantly impact the results.
10 Requirements

As the purpose of the simulation is to demonstrate that the Attitude Determination and Control Subsystem meets its requirements. Since the simulation is not concerned with the hardware or interface software requirements only the system performance requirements are listed below

- The subsystem shall achieve an angular rate $< 0.20^\circ/s$ within fifteen minutes of being powered on, $0.20^\circ/s$ is 80% of the rate limit for the star tracker.
- The subsystem shall point the solar array normal vector to within $14^\circ$ of the sun vector during normal operations, this corresponds to a 3% loss in power generation.
- The subsystem shall point the transmit antenna boresight to within $15^\circ$ of the ground station vector during S-Band communications, this corresponds to half of the half-power beamwidth of the antenna for $\sim$1dB of loss.
- The subsystem shall point the mirror boresight to within $2.3^\circ$ of nadir during science operations, this is the beamwidth of the science instrument.
- The subsystem shall determine the attitude of the satellite to within $0.23^\circ$ during science operations, this is one-tenth of the beamwidth of the science instrument.

There is not a specific attitude knowledge requirement for the non-science modes as the attitude only needs to be known well enough to allow the pointing requirements to be met.

11 Configuration

11.1 Control Gains

The main tuning parameter for the control design is the control gains for each mode. The gains are the attitude error gain $K$, the rate error gain matrix $[P]$, the integrated attitude error gain matrix $[K_I]$, and the momentum management gain $K_M$. The gains and tuning logic for each mode are summarized in the following sections.

11.1.1 Detumble

As the purpose of detumble is simply to bring the satellite rates down to below the threshold rate for achieving a star camera solution the attitude gain and integrated attitude gains are both set to zero. Since the star tracker cannot provide an attitude solution with which to compute the attitude error until detumble is complete there is no use for those gains in any case. Since the angular rates do not need to be brought down particularly quickly, and the wheel speeds need to be brought down before disabling the control while initializing the attitude determination filter, the rate error gain is set to $[P] = diag([0.001, 0.001, 0.001])$ and the momentum management gain is set to $K_M = 0.0025$.

11.1.2 Sun-Pointing

Since the majority of the mission duration is spent in sun-pointing mode one of the design considerations for this mode is low control torque. As the relative position of the sun is only changing at the revolution rate of the Earth holding the satellite pointed at the sun requires very little control so the gains can be selected for the mode transitions. An attitude error gain of $K = 0.005$ and rate error gain matrix of $[P] = diag([0.0075, 0.0075, 0.0075])$ provides smooth mode transitions and very good steady-state error. The integrated attitude error gain is still set to zero and the momentum management gain remains $K_M = 0.0025$.

11.1.3 S-Band Communications

As the requirements for the S-Band Communication mode are very similar to those for sun-pointing, and the required angular rates to maintain tracking of the ground station are not high, the gains for this mode are set the same as for sun-pointing.
11.1.4 Science

Since the pointing requirements for science mode are much tighter than for the other modes and there is then the disturbance of the mirror the gains are set higher than for the previous modes. The attitude error gain is increased to $K = 0.0075$ and the rate error gain is increased to $\begin{bmatrix} P \end{bmatrix} = \text{diag}(0.01, 0.01, 0.0125)$. The rate error gain for the third axis is set higher as the primary effect of the mirror disturbance is to produce a rate about the long axis of the satellite, due both to the orientation of the mirror rotation axis and the smaller satellite inertia about that axis. Science mode also includes an integrated attitude error gain of $\begin{bmatrix} K_I \end{bmatrix} = \text{diag}(0.0025, 0.0025, 0.0025)$ to help clean up external disturbances. The momentum management gain is set to zero in this mode to avoid any attitude errors caused by inaccuracies in the magnetic field measurement, as the reaction wheel correction is based on the assumed torque generated by the torque rods, and to alleviate concerns of EMI between the torque rods and science instrument.

11.2 Extended Kalman Filter

The tuning parameter for the EKF is the process noise covariance matrix $Q$. As the process noise covariance comes into the filter in the time update of the state covariance matrix, as an inflation of the covariance, a larger process noise covariance increases the uncertainty in the state estimate. This has the effect of weighting the measurements more strongly relative to the propagated state in the computation of the Kalman gain. This avoids filter saturation, where the state covariance becomes so small that measurements are effectively ignored, but can potentially result in over-fitting the measurements. As the measurements include some noise an excessively large process noise covariance results in the filter chasing measurement noise. The process noise covariance matrix was experimentally set, by scaling a matrix of the gyroscope noise and estimated bias variability, to $Q = \text{diag}(8.11 \times 10^{-8}, 8.11 \times 10^{-8}, 8.11 \times 10^{-8}, 8.11 \times 10^{-14}, 8.11 \times 10^{-14}, 8.11 \times 10^{-14})$.

11.3 Gyroscope Filtering

The gyroscope filter is configured by the number of measurements used in the sliding-average filter. As the non-science modes do not require fine attitude knowledge, and we would like to reduce the control effort required in those modes to prolong the subsystem lifetime, the averaging interval is 25 points (0.25 seconds). In science mode the averaging interval is reduced to four points (0.04 seconds) to reduce the impact of the filter delay. As only a small fraction of the mission is spent in science mode the additional control torque required is not a concern.

11.4 Control Update Rate

The obvious tuning parameter for the control as a whole is the rate at which new control outputs are computed. The considerations are that if the control is updated too slowly then any higher-frequency effects will not be compensated for and the satellite could drift significantly between updates but the control can only be updated at a certain rate, due to hardware limitations, and a high update rate could excite flexible structural modes. A high update rate may also cause the control to respond to high-frequency but small-amplitude zero-mean effects, increasing the control effort for no improvement in pointing accuracy. The control update rate was experimentally set to 4Hz to provide a sufficient update rate to maintain tracking while avoiding phasing interactions with the mirror rotation as well as introducing high-frequency control inputs that might excite flexible structural modes.

11.5 Control Smoothing

Since there is a reduced averaging time for the gyroscope measurements in science mode some smoothing of the control output is used to reduce the control effort. This is implemented by simply setting the actual control output to the average of the three most recent computed torques. As the control is evaluated at 4Hz this is an averaging time of 0.75 seconds. The rotation period of the mirror is 0.76 seconds so this is the average control response over one full rotation of the mirror. This allows small variations from the current position of the mirror to average out so the control is only addressing the secular variations while avoiding phasing-interaction concerns from simply setting the control to a slower update rate relative to the mirror rotation rate.
12 Simulation Verification

Before using the simulation to assess performance requirements it is critical to first ensure that the simulation is internally consistent. If all control is disabled then the simulation should conserve both angular momentum and kinetic energy and if the reaction wheels, but not the torque rods, are enabled then the simulation should still conserve angular momentum and the rate of kinetic energy change should match the known power to the reaction wheels. This testing does not validate that the simulation is accurately modeling PolarCube, as that would require comparison of simulation output with test data, but it does verify that the simulation is internally consistent. The variations are assessed as the relative error:

$$\epsilon_{rel} = \frac{X_{sim} - X_{ref}}{X_{ref}}$$ (18)

12.1 Natural Dynamics

This is the case where all control is disabled. The simulation is initialized with some satellite angular rates and wheel speeds and then these rates are allowed to evolve according to the natural dynamics of the system. The plots of angular momentum and kinetic energy, over a section of the sun-pointing mode so the mirror is not rotating, for this case are shown in Figure (5) and Figure (6).

![Figure 5: Angular Momentum Variation](image)

![Figure 6: Kinetic Energy Variation](image)

The variation in both values is roughly 0.3% in 400 seconds, which is 2.7% per simulation hour. This is considered to be an acceptable variation as this indicates that there are no implementation errors and the drift is simply due to the use of a simple Euler integrator and accumulated integration error. Increasing the simulation frequency and/or replacing the Euler integrator would improve this issue but would increase the execution time of the simulation.

12.2 Natural Dynamics plus Mirror

As there is not torque-level simulation of the mirror in the current simulation the case where the mirror is allowed to rotate naturally is not possible. This means that kinetic energy conservation cannot be verified with the mirror rotating as there is an implied motor torque required to maintain the specified rate. The angular momentum for this case is shown in Figure (7).
The same drift that was seen in the previous case is still present with some additional high-frequency variations due to the rotation of the mirror. As the overall drift with the mirror rotating is the same as without the mirror rotating the mirror is consistently handled in the simulation.

### 12.3 Reaction Wheels

When the reaction wheels are enabled the simulation will no longer conserve kinetic energy, though the rate of kinetic energy change should then match the power for the reaction wheels, and angular momentum will still be conserved. The power is computed using Equation (19).

\[
P = \omega^\top L + \sum_{i=1}^{3} \Omega_i u_{s_i}
\]  

(19)

The torque rods are still disabled as they produce an external torque so angular momentum would not be conserved. The plots of angular momentum and power/kinetic energy rate for this case are shown in Figure (8) and Figure (9).

As there is torque-level simulation of the reaction wheels in the simulation the fact that the drift with the
control enabled is the same as without is as expected. The power/kinetic energy rate plot shows that the change in system kinetic energy is consistent with the power equation, showing that the system dynamics are consistent.

12.4 Full Control

With the torque rods enabled the power equation can still be used to compare the rate of kinetic energy change to the power, now inclusive of the external torque. The power/kinetic energy rate plot for this case is shown in Figure (10).

![Power and Kinetic Energy Rate](image)

Figure 10: Power and Kinetic Energy Rate

This plot both shows that the power equation results and numerical kinetic energy rate remain consistent with the addition of the torque rods and that the overall power is lower. This is since the power is dependent on the reaction wheel speeds, so the same control torque at lower reaction wheel speeds requires less power, and the momentum management control reduces the wheel speeds.

13 Major Results

The simulation is run for 45 minutes and the simulation start time was selected so that there is one full run of all modes in this time. This allows the design to be fully assessed without waiting longer than needed for the results. The orbit used in the simulation is inclined at 80°, has a perigee altitude of 400km, and an apogee altitude of 420km. This is a reasonable test orbit for the desired science data products. A stripped-down version of the simulation that consists of just the orbit propagator and mode checks is used to select the simulation start time. As this version can be run at a much coarser time step than the main simulation, and involves a small fraction of the computations, several days of simulation time can be quickly assessed to select a simulation period that shows all modes.

As the requirements are specified by mode breaking up the results by mode provides the easiest assessment of the simulation results. The results for assessing the actuator limits and filter performance are also addressed in their respective sections.

13.1 Detumble

The principal concerns for detumble are that ADCS achieves the specified rates within the required time and that the gyroscope bias estimate from the attitude determination filter converges to the true value. The angular rates during detumble are shown in Figure (11) with the angular rate requirement marked with the black lines.
The rates very quickly converge to within the requirements though do not go to zero. This is because the initial gyroscope bias estimate is not the true value and the control is driving the estimated rates to zero. After ten minutes, to allow the initial detumble to converge and stabilize and the momentum management to bring the wheel speeds back down, the system disables the control and starts the EKF. After ten star tracker solutions the control is re-enabled. The control is disabled initially to prevent a jump in the estimated gyroscope bias from causing the control to drive the rates above the threshold for a star tracker solution. If this happens then the satellite cannot achieve further star tracker solutions to fix the bias estimate. The wheels speeds need to be relatively low during this period as well to prevent the reaction wheel gyroscopic torques generated by the residual angular rates from causing the rates to diverge. Once the control is re-enabled the rates converge to zero as the gyroscope bias estimate is now accurate so the estimated rates match the true rates.

13.2 Sun-Pointing

The requirement for the sun-pointing mode is that ADCS will orient the solar panel normal vector to within $14^\circ$ of the sun vector. The pointing error during this mode is shown in Figure (12) with the pointing error marked with the black line.
The attitude error is initially outside the requirement, as the satellite has to re-orient from the previous pointing target, but quickly converges to well within the requirement. The half-minute required to converge to within the requirement is a very small fraction of the total time in the sun-pointing mode so is not a concern from a system power standpoint.

13.3 S-Band Communications

The requirement for the S-Band Communications mode is that ADCS will orient the transmit antenna boresight to within $15^\circ$ of the vector to the grounds station. The pointing error during this mode is shown in Figure (13).

![Figure 13: Pointing Error](image)

As with the sun-pointing mode the attitude error is initially outside the requirement, as the satellite has to re-orient from the previous pointing target, but subsequently converges to well within the requirement. This concern is addressed by commanding the subsystem to begin tracking the ground station before the ground station is actually in view so the antenna is correctly pointing when communications start.

13.4 Science

The requirements for the science mode are that the mirror beam be pointed within $2.3^\circ$ of nadir and that the attitude be known to within $0.23^\circ$. The pointing error during the science data collection mode is shown in Figure (14) and the determination error is shown in Figure (15) with the requirements again marked with the black lines.
The subsystem takes a little over a minute to re-orient to the pointing target for science mode and then holds the attitude well inside the requirement for the remainder of the mode. Similar to the S-Band Communications mode the time to re-orient is addressed by commanding the new pointing target before the science instrument begins collecting data so by the time that data collection starts the satellite is oriented correctly.

As the science mode also has a specific attitude knowledge requirement, that is tighter than what is necessary to just meet the pointing error requirement, the knowledge error needs to be independently assessed. The knowledge error is shown to consistently meet the requirement. The primary source of knowledge error is the noise in the gyroscopes, even after smoothing the gyroscope output before the main filter, so if further improvement in the attitude knowledge accuracy is required then the direct output of the gyroscopes will need to be improved.

### 13.5 Actuator Limits

The actuator limits are more clearly assessed over the full mode sequence as they are not tied to specific modes. The reaction wheel control torques over the full simulation duration are shown in Figure (16) and the wheel speeds are shown in Figure (17).

The reaction wheel torques are limited by the simulation to the specified limit of 0.27mNm, the torques saturate during mode transitions while the satellite re-orient for the new target but are otherwise very small. The
highest reaction wheel speeds also occur during mode transitions though as a maneuver causes no net change in
the wheel speeds this spike is solely determined by the angular rate during the maneuver. Executing the same
maneuver more slowly requires less momentum to be transferred to the reaction wheels and keeps their speeds
down. In the event that the torques or wheel speeds seen during mode transitions are unacceptable, due to new
test data or lifetime considerations, a transition mode could be introduced where the gains are set lower during
the re-orientation than they are to meet the pointing requirements during the mode. The simulation includes the
structure for this transition mode but it is currently disabled. The other concern for the wheel speeds is that for
the initial detumble all of the satellite momentum is transferred to the wheels and then must be dissipated using
the torque rods. The initial slow rise of the wheel speeds from 0-250 seconds is the initial detumble and the slow
decay of wheel speeds until re-orienting for sun-pointing at 900 seconds is this momentum management.

13.6 Filter Performance

A critical assessment of the attitude determination filter performance is to verify that the state errors do fall within
the $3\sigma$ covariance envelopes. This check verifies that the reported uncertainty in the state estimate is consistent
with the actual accuracy of the state estimate. The state errors and covariance envelopes for the attitude estimate
and gyroscope bias estimate are shown in Figure (18) and Figure (19) respectively.

![Figure 18: MRP Errors and Covariance](image)

![Figure 19: Gyroscope Bias Errors and Covariance](image)

The state error/covariance plots show that the reported covariance is an appropriate representation of the
accuracy of the state estimate.

Another important check for the filter is the post-fit measurement residuals. These, along with the $3\sigma$ bounds
for the known star tracker measurement noise, are shown in Figure (20).

19
The post-fit residuals should be consistent with the known measurement noise as that shows that the filter is appropriately accounting for the measurements, neither ignoring measurements nor over-fitting measurement noise, as they are in the plot above. This result, along with the state errors/covariance shown previously, shows that the filter is performing as it should and that the process noise covariance has been correctly chosen. The apparent signals in the residuals are most likely due to the effects of the gyroscope noise as the state estimate is updated more often than star camera solutions are available. The gaps in the post-fit residuals are the periods where the satellite angular velocity exceeded the rate limit for the star tracker.

14 Summary

The current simulation shows that the ADCS design, given the current assumptions and test data, meets its requirements. The simulation structure is designed to make it easier for a new student on the team to learn how to use the simulation and continue assessing the ADCS design as well as allowing for easy update of any modules. The current ADCS simulation provides a high-fidelity and user-friendly means of assessing the performance of the subsystem and the effects of possible mission design changes.

References


Appendix A: Equations of Motion

The equations of motion for a rigid body, or system of rigid bodies, are given by Euler’s Equation, where $\dot{H}$ is the inertial derivative of the system angular momentum and $L$ are the external torques acting on the system:

$$\dot{H} = L$$

The angular momentum of the satellite with the mirror and one reaction wheel is defined by:

$$H = H_B + H_M + H_W$$

The following subsections cover the angular momentum and its inertial derivative for the satellite body, a single reaction wheel, and the mirror then combine those results to give the equations of motion for the full system of satellite body, mirror, and three reaction wheels.

A.1: Satellite Body

The angular momentum of the satellite body is given by:

$$B H_B = B [I_S] \omega_{B/N}$$

Simplifying notation as $\omega$, with no subscripts to assign explicit frames, is understood to mean $\omega_{B/N}$:

$$B H_B = B [I_S] \omega$$

Taking the inertial derivative and dropping the explicit frame assignments:

$$\dot{H}_B = \frac{N}{dt} [I_S] \omega$$

Expanding the inertial derivative using the transport theorem, the notation $[\omega]$, with no subscript, is used for what is explicitly $[\omega_{B/N}]$:

$$\dot{H}_B = \frac{B}{dt} [I_S] \omega + [\omega] [I_S] \omega$$

As $[I_S]$ is constant in the body frame the final form is:

$$\dot{H}_B = [I_S] \dot{\omega} + [\omega] [I_S] \omega$$

A.2: Reaction Wheel

The angular momentum of a rigid single wheel is given by:

$$B H_W = B [I_W] \omega_{W/B}$$

Rotating the wheel inertia from its known values in wheel-frame to body-frame is accomplished by pre-multiplying by the rotation matrix from wheel-frame to body-frame and post-multiplying by the transpose of the rotation matrix.

$$B [I_W] = [BW] [I_W] [BW]^T; \quad [BW] = [\hat{w}_t, \hat{w}_t, \hat{w}_s]$$

Substituting this result along with $\omega_{W/B} = \omega_{W/B} + \omega_{B/N}$ and then $\omega_{W/B} = \Omega \hat{w}_s$ into the expression for $H_W$ gives:

$$B H_W = \left( [BW] [I_W] [BW]^T \right) \omega_{W/B} + J_s \Omega \hat{w}_s$$

As the wheel inertia tensor $W [I_W]$ is diagonal the term in parentheses can be simplified to give:

$$B H_W = (J_t \hat{w}_t \hat{w}_t^T + J_t \hat{w}_t \hat{w}_t^T + J_s \hat{w}_s \hat{w}_s^T) \omega_{W/B} + J_s \Omega \hat{w}_s$$
Defining \( \omega_{t_1} = \mathbf{w}_{t_1}^T \omega_{B/N} \), \( \omega_{t_2} = \mathbf{w}_{t_2}^T \omega_{B/N} \), and \( \omega_s = \mathbf{w}_s^T \omega_{B/N} \) then regrouping terms gives the final form of:

\[
\mathbf{B} \mathbf{H}_W = J_t \omega_{t_1} \mathbf{w}_{t_1} + J_t \omega_{t_2} \mathbf{w}_{t_2} + J_s (\omega_s + \Omega) \mathbf{w}_s
\]

Taking the inertial derivative and dropping the explicit frame assignments:

\[
\dot{\mathbf{H}}_W = \frac{N}{d} [J_t \omega_{t_1} \mathbf{w}_{t_1}] + \frac{N}{d} [J_t \omega_{t_2} \mathbf{w}_{t_2}] + \frac{N}{d} [J_s (\omega_s + \Omega) \mathbf{w}_s]
\]

Pulling the \( J_x \) through the derivative operators as they are constants:

\[
\dot{\mathbf{H}}_W = J_t \frac{N}{d} [\omega_{t_1} \mathbf{w}_{t_1}] + J_t \frac{N}{d} [\omega_{t_2} \mathbf{w}_{t_2}] + J_s \frac{N}{d} [(\omega_s + \Omega) \mathbf{w}_s]
\]

Expanding the inertial derivative using the transport theorem, the notation \([\bar{\omega}]\), with no subscript, is used for what is explicitly \([\bar{\omega}_{B/N}]\):

\[
\dot{\mathbf{H}}_W = J_t \left( \frac{B}{d} \frac{d}{dt} [\omega_{t_1} \mathbf{w}_{t_1}] + [\bar{\omega}] \omega_{t_1} \mathbf{w}_{t_1} \right) + J_t \left( \frac{B}{d} \frac{d}{dt} [\omega_{t_2} \mathbf{w}_{t_2}] + [\bar{\omega}] \omega_{t_2} \mathbf{w}_{t_2} \right) + J_s \left( \frac{B}{d} \frac{d}{dt} [(\omega_s + \Omega) \mathbf{w}_s] + [\bar{\omega}] (\omega_s + \Omega) \mathbf{w}_s \right)
\]

To evaluate these derivatives we first need the inertial derivatives of the velocity projections \( \omega_{t_1} \), \( \omega_{t_2} \), and \( \omega_s \), these are computed from the definitions of the velocity projections:

\[
\frac{B}{d} \frac{d}{dt} [\omega_{t_1}] = \mathbf{w}_{t_1}^T \omega_{B/N} \quad \frac{B}{d} \frac{d}{dt} [\omega_{t_2}] = \mathbf{w}_{t_2}^T \omega_{B/N} \quad \frac{B}{d} \frac{d}{dt} [\omega_s] = \mathbf{w}_s^T \omega_{B/N}
\]

As the wheel frame is fixed in the body frame, as the transverse axes can be held fixed due to the rotational symmetry of the wheel, the body-frame derivatives of the wheel-frame unit vectors are zero. This simplifies the derivatives to:

\[
\frac{B}{d} \frac{d}{dt} [\omega_{t_1}] = \mathbf{w}_{t_1}^T \omega_{B/N} \quad \frac{B}{d} \frac{d}{dt} [\omega_{t_2}] = \mathbf{w}_{t_2}^T \omega_{B/N} \quad \frac{B}{d} \frac{d}{dt} [\omega_s] = \mathbf{w}_s^T \omega_{B/N}
\]

Substituting these results into the previous expression for \( \dot{\mathbf{H}}_W \) gives the final form for a single wheel of:

\[
\mathbf{H}_W = J_t \left( \mathbf{w}_{t_1} \mathbf{w}_{t_1}^T \dot{\omega} + [\bar{\omega}] \omega_{t_1} \mathbf{w}_{t_1} \right) + J_t \left( \mathbf{w}_{t_2} \mathbf{w}_{t_2}^T \dot{\omega} + [\bar{\omega}] \omega_{t_2} \mathbf{w}_{t_2} \right) + J_s \left( \left( \mathbf{w}_s^T \dot{\omega} + \Omega \right) \mathbf{w}_s + [\bar{\omega}] (\omega_s + \Omega) \mathbf{w}_s \right)
\]

**A.3: Mirror**

The second term is the angular momentum of the mirror, this is given by:

\[
\mathbf{B} \mathbf{H}_M = \mathbf{B} [I_M] \omega_{M/N}
\]

Rotating the mirror inertia from its known values in mirror-frame to body-frame is accomplished by pre-multiplying by the rotation matrix from mirror-frame to body-frame and post-multiplying by the transpose of the rotation matrix.

\[
\mathbf{B} [I_M] = [BM] M [I_M] [BM]^T \quad [BM] = [\mathbf{m}_s \mathbf{m}_t \mathbf{m}_r]
\]

Substituting this result in along with \( \omega_{M/N} = \omega_{M/B} + \omega_{B/N} \) and then \( \omega_{M/B} = \mu \mathbf{m}_r \) gives:

\[
\mathbf{B} \mathbf{H}_M = \left( [BM] M [I_M] [BM]^T \right) (\omega_{B/N} + \mu \mathbf{m}_r)
\]

As \( M [I_M] \) is fully populated the axes are not easily separated so this is the form used.

Simplifying notation as \( \omega \), with no subscripts to assign explicit frames, is understood to mean \( \omega_{B/N} \):

\[
\mathbf{B} \mathbf{H}_M = \left( [BM] M [I_M] [BM]^T \right) (\omega + \mu \mathbf{m}_r)
\]
The angular momentum of the full satellite is the sum of the previous results, the final form is:

$$^{B}\mathbf{H} = ^{B}[s] \omega_{B/N} + (BM)^{M}[I_{M}] [BM]^{T} (\omega + \dot{\mu}m_{r})$$

Expanding the derivative using the product rule:

$$\dot{H}_{M} = \frac{d}{dt} \left( (BM)^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r})$$

Expanding the inertial derivatives using the transport theorem, the notation $[\bar{\omega}]$, with no subscript, is used for what is explicitly $[\bar{\omega}_{B/N}]$:

$$\dot{H}_{M} = \left( \frac{d}{dt} \left( [BM]^{M}[I_{M}] [BM]^{T} \right) + [\bar{\omega}_{M/N}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) \right) (\omega + \dot{\mu}m_{r})$$

$$+ \left( [BM]^{M}[I_{M}] [BM]^{T} \right) \left( \frac{d}{dt} (\omega + \dot{\mu}m_{r}) + [\bar{\omega}] (\omega + \dot{\mu}m_{r}) \right)$$

Evaluating the body-frame derivative, using the fact that $m_r$ is fixed in the body frame:

$$\dot{H}_{M} = \left( \frac{d}{dt} \left( [BM]^{M}[I_{M}] [BM]^{T} \right) + [\bar{\omega}_{M/N}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) \right) (\omega + \dot{\mu}m_{r})$$

$$+ \left( [BM]^{M}[I_{M}] [BM]^{T} \right) ((\omega + \dot{\mu}m_{r}) + [\bar{\omega}] (\omega + \dot{\mu}m_{r}))$$

As $[M[I_{M}]$ is constant in the mirror frame and the columns of $[BM]$ are the mirror-frame unit vectors, which are fixed in the mirror frame, the first term drops out:

$$\dot{H}_{M} = [\bar{\omega}_{M/N}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r})$$

$$+ \left( [BM]^{M}[I_{M}] [BM]^{T} \right) ((\omega + \dot{\mu}m_{r}) + [\bar{\omega}] (\omega + \dot{\mu}m_{r}))$$

Separating $[\bar{\omega}_{M/N}]$ into $[\bar{\omega}_{M/B}] + [\bar{\omega}]$:

$$\dot{H}_{M} = [\bar{\omega}_{M/B}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r}) + [\bar{\omega}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r})$$

$$+ \left( [BM]^{M}[I_{M}] [BM]^{T} \right) ((\omega + \dot{\mu}m_{r}) + [\bar{\omega}] (\omega + \dot{\mu}m_{r}))$$

Simplifying using $[\bar{\omega}] \omega = 0$:

$$\dot{H}_{M} = [\bar{\omega}_{M/B}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r}) + [\bar{\omega}] \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r})$$

$$+ \left( [BM]^{M}[I_{M}] [BM]^{T} \right) (\omega + \dot{\mu}m_{r} + [\bar{\omega}] \dot{\mu}m_{r})$$

A.4: Full Satellite

The angular momentum of the full satellite is the sum of the previous results, the final form is:

$$^{B}\mathbf{H} = ^{B}[s] \omega_{B/N} + (BM)^{M}[I_{M}] [BM]^{T} (\omega_{B/N} + \dot{\mu}m_{r})$$

$$+ \sum_{i=1}^{3} J_{i} \omega_{t_{i}} \dot{\omega}_{t_{i}} + J_{i} \omega_{t_{i}} \dot{\omega}_{t_{i}} + J_{s_{i}} (\omega_{s_{i}} + \Omega_{i}) \dot{\omega}_{s_{i}}$$

Since the angular momentum derivatives have been already been separately computed for each section of the system the expressions can be substituted into Euler’s Equation to determine the final equations of motion for

23
the satellite. The separate derivatives are:

\[
\dot{H}_B = [I_S] \dot{\omega} + [\omega] [I_S] \omega
\]

\[
\dot{H}_W = J_t (\tilde{w}_{t_1} \tilde{w}_{t_1}^T \omega + [\omega] \omega_{t_1} \tilde{w}_{t_1}) + J_t (\tilde{w}_{t_2} \tilde{w}_{t_2}^T \omega + [\omega] \omega_{t_2} \tilde{w}_{t_2}) + J_s \left( (\tilde{w}_s^T \omega + \Omega) \tilde{w}_s + [\omega] (\omega_s + \Omega) \tilde{w}_s \right)
\]

\[
\dot{H}_M = [\omega_{M/B}] \left( [BM] M [IM] [BM]^T \right) (\omega + \mu \tilde{m}_r) + [\omega] \left( [BM] M [IM] [BM]^T \right) (\omega + \mu \tilde{m}_r) + \left( [BM] M [IM] [BM]^T \right) (\omega + \mu \tilde{m}_r + [\omega] \mu \tilde{m}_r)
\]

Substituting these expressions into Euler’s Equation gives:

\[
H = H_S + H_W + H_M = L
\]

\[
[I_S] \dot{\omega} + [\omega] [I_S] \omega + J_t (\tilde{w}_{t_1} \tilde{w}_{t_1}^T \omega + [\omega] \omega_{t_1} \tilde{w}_{t_1}) + J_t (\tilde{w}_{t_2} \tilde{w}_{t_2}^T \omega + [\omega] \omega_{t_2} \tilde{w}_{t_2}) + J_s \left( (\tilde{w}_s^T \omega + \Omega) \tilde{w}_s + [\omega] (\omega_s + \Omega) \tilde{w}_s \right)
\]

Rearranging so that [I] \dot{\omega} is isolated on one side of the equation gives the standard form:

\[
[I] \dot{\omega} = -[\omega] [I] \omega - J_s \left( \tilde{\Omega} \tilde{w}_s + [\omega] \Omega \tilde{w}_s \right) - [\omega_{M/B}] \left( [BM] M [IM] [BM]^T \right) (\omega + \mu \tilde{m}_r) - [\omega] \left( [BM] M [IM] [BM]^T \right) (\mu \tilde{m}_r) - [BM] M [IM] [BM]^T (\mu \tilde{m}_r + [\omega] \mu \tilde{m}_r) + L
\]

This expression is for a satellite for a single reaction wheel so the equation must still be expanded to include the three wheels of the spacecraft. This is simply done by summing the reaction wheel terms over the number of reaction wheels to produce the following expression:

\[
[I] \dot{\omega} = -[\omega] [I] \omega - [\omega_{M/B}] \left( [BM] M [IM] [BM]^T \right) (\omega + \mu \tilde{m}_r) - [\omega] \left( [BM] M [IM] [BM]^T \right) (\mu \tilde{m}_r) - [BM] M [IM] [BM]^T (\mu \tilde{m}_r + [\omega] \mu \tilde{m}_r) + L
\]

Where the full inertia [I] is now defined as:

\[
[I] = [I_S] + [BM] M [IM] [BM]^T + \sum_{i=1}^3 J_t_i \tilde{w}_{t_i} \tilde{w}_{t_i}^T + J_s \tilde{w}_s \tilde{w}_s^T
\]

Appendix B: Extended Kalman Filter

The general structure of the sequential filter was discussed in the body of the paper so this appendix will only go through the specific equations for the application of the EKF to attitude determination using MRPs.

Most of the following developments are summarized from O’Keefe and Schaub [3] and Karlaard and Schaub [4].
B.1: EKF Derivations for MRPs

The implementation of the EKF requires the equations for the state vector derivative, the matrices of first-order partial derivatives, and the mapping of the state estimate and covariance matrix through the MRP set switches. The following derivations are simply a summary of the O’Keefe and Schaub paper [3] with the sole difference that the gyroscope measurement is a composite of the two gyroscopes, this is only a conceptual difference and does not affect the math. Before going into the derivations it is first important to define the terms and notation used.

\( x \): State Vector
\( P \): Variance-Covariance Matrix, commonly referred to as simply the Covariance Matrix
\( x^\perp_k \): State vector mapped to current time \( k \) but not updated for new measurements
\( x^+ k \): State vector at time \( k \) after measurement update
\( \hat{P}^\perp_k \): Covariance matrix mapped to current time \( k \) but not updated for new measurements
\( \hat{P}^+ k \): Covariance matrix at time \( k \) after measurement update
\( K \): Kalman Gain
\( y \): Measurement Residual
\( R \): Observation Error Covariance Matrix
\( Q \): Process Noise Covariance Matrix

The two gyroscopes are averaged together to reduce noise before calling the filter, the filter is then estimating the bias of this composite sensor. The state vector for the attitude and composite bias is \( x = [\sigma \omega b]^\top \), the state dynamics are then:

\[
\dot{x} = f(x) + g(x, \eta)
\]  
(20)

Where the \( f \) and \( g \) functions are defined as:

\[
f(x) = \left[ \frac{1}{2} B(\sigma) \omega \right]
\]  
(21)

\[
g(x, \eta) = \left[ -\frac{1}{4} B(\sigma) \eta \omega \right]
\]  
(22)

As the covariance update in the EKF uses the linearized system dynamics the matrices of first-order partial derivatives of the \( f \) and \( g \) functions with respect to states are required, these are:

\[
F \equiv \frac{\partial f}{\partial x} \bigg|_{x=x^\perp} = \left[ \frac{1}{2} \left( \sigma \dot{\omega} - \omega \dot{\sigma}^\top - [\dot{\omega}]_{3 \times 3} + \dot{\sigma}^\top \omega I_{3 \times 3} \right) - \frac{1}{4} B(\sigma) \right]
\]  
(23)

\[
G \equiv \frac{\partial g}{\partial x} \bigg|_{x=x^\perp, \eta=0} = \left[ -\frac{1}{4} B(\sigma) \right]
\]  
(24)

The differential equation for the state covariance is then:

\[
\dot{\hat{P}} = F \hat{P}^\top + \hat{P} F^\top + G Q G^\top
\]  
(25)

A critical consideration when working with MRPs is how the state and state covariance are propagated through the set switches. The state mapping is simply the usual shadow set mapping for the MRPs for the \( \sigma \) component of the state vector and the bias components are left as is. Written out for the full state vector this is:

\[
x^S = \lambda(x) = \left[ \begin{array}{c} -\sigma \sigma \omega \omega \end{array} \right]
\]  
(26)

The mapping of the covariance matrix is more complicated but can be simplified by breaking the covariance matrix into submatrices for the \( 3 \times 3 \) covariance matrices for each component of the state vector and the \( 3 \times 3 \)
cross-correlation matrices between components. This looks like:

\[ \hat{P} = \begin{bmatrix} P_{\sigma \sigma} & P_{\sigma \omega_b} \\ P_{\omega_b \sigma} & P_{\omega_b \omega_b} \end{bmatrix} \]  

(27)

Following the work of Karlgaard and Schaub [4] the shadow-set covariance matrix is given by:

\[ \hat{P}^S = \Lambda \hat{P} \Lambda^T = \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{\sigma \sigma} & P_{\sigma \omega_b} \\ P_{\omega_b \sigma} & P_{\omega_b \omega_b} \end{bmatrix} \begin{bmatrix} \Lambda_{11}^T & 0 \\ 0 & I \end{bmatrix} \]  

(28)

Multiplying out the matrix products gives:

\[ \hat{P}^S = \begin{bmatrix} \Lambda_{11} P_{\sigma \sigma} \Lambda_{11}^T & \Lambda_{11} P_{\sigma \omega_b} \\ P_{\omega_b \sigma} \Lambda_{11}^T & P_{\omega_b \omega_b} \end{bmatrix} \]  

(29)

The form of \( \Lambda \) is still needed, this is computed by:

\[ \Lambda = \frac{\partial \lambda}{\partial x} = \begin{bmatrix} 2\sigma^{-4}\sigma\sigma^T - \sigma^{-2}I & 0 \\ 0 & I \end{bmatrix} \]  

(30)

As only the upper-right sub-matrix is needed for the covariance mapping this is pulled out as \( \Lambda_{11} = 2\sigma^{-4}\sigma\sigma^T - \sigma^{-2}I \). The non-bold \( \sigma \) in the equations is the scalar magnitude of the MRP, computed as \( \sigma = \sqrt{\sigma^T\sigma} \).

The final piece is the measurement equation. This describes the expected measurement in terms of the state variables. As the star tracker directly measures the current attitude this equation is simply:

\[ h(x) = \hat{\sigma} \]  

(31)

The filter uses the Jacobian of this equation with respect to states evaluated on the reference, this is:

\[ H_k \equiv \frac{\partial h}{\partial x} \bigg|_{\hat{x}_k} = [I_{3 \times 3} \ 0_{3 \times 3}] \]  

(32)

The mapping of the state estimate and covariance matrix through the set switch has been discussed but there is also the question of how to handle the measurement residual \( y_k = \tilde{\sigma}_k - \hat{\sigma}_k \). The O’Keefe paper [3] goes into more detail and discusses the rationale for the threshold values but the results are summarized here. The question for the measurements is which MRP set should be used to represent the measurement. If the measured MRP is reported in a different set than the current attitude estimate the difference will be large, even if the actual difference is quite small. The best illustration of this issue is that the MRPs \([1 \ 0 \ 0]^T\) and \([-1 \ 0 \ 0]^T\) are actually the same orientation but the measurement residual would be \( y_k = [2 \ 0 \ 0]^T \). This is handled by computing the measurement residual using the measurement represented in both sets and then using the residual with the smaller norm. As the shadow-set representation is not defined at if the primary set norm is zero, and the shadow-set result will clearly be larger than the primary set result near zero, the shadow-set result is only checked if the norm of the measured MRP exceeds a threshold value, chosen to be \( \frac{1}{3} \). The pseudocode for computing the measurement residual is then:

1. \( y_k = \tilde{\sigma}_k - \hat{\sigma}_k \)
2. if \( ||\tilde{\sigma}_k|| > \frac{1}{3} \) then
3. \( y'_k = \tilde{\sigma}_k - \hat{\sigma}_k \)
4. if \( ||y'_k|| < ||y_k|| \) then
5. \( y_k = y'_k \)
6. end if
7. end if
B.2: Implementation of the EKF

Now that all of the MRP-specific derivations for the EKF have been completed the general process for the EKF and the implementation details can be discussed. The EKF is composed of two major steps; the time update and the measurement update.

1. Time Update
   (a) Propagate the state estimate and covariance matrix to current time by integrating Eq. (21) and Eq. (25).

2. Measurement Update
   (a) Compute the Kalman Gain
       \[ K_k = \hat{P}_k^{-1/2} H^T \left[ H \hat{P}_k^{-1/2} H^T + R \right]^{-1} \]
   (b) Compute the measurement residual \( y_k \), switch sets as appropriate using the method discussed in the derivations
   (c) Update the state estimate \( \hat{x}_k^+ = \hat{x}_k^- + K_k y_k \)
   (d) Update the covariance matrix \( \hat{P}_k^+ = [I - K_k H] \hat{P}_k^- [I - K_k H]^T + K_k R K_k^T \)

In the measurement update equations the explicit time subscript \( k \) has been dropped for \( H \) and \( R \) as these are constants in this case. The form of the covariance update is the Joseph Formulation which improves numeric stability for the filter.