Cornell University SIOS Lab

Masters of Engineering Final Report

Advisor: Dmitry Savransky

Geometry Cross-Calibration of Orbiting Satellite Swarms

Authors:

Thomas Taffe
Van Cates

December 23, 2021

Contents

1 Abstract

Summary of research on Dovesat parameters, Attitude Control System (ACS), ANSYS transient analysis, and Measurement Function.

2 Introduction

By using multiple small satellites (CubeSats) that make up a "constellation", the objective of the CubeSat constellation is to produce high-resolution images of any celestial body. This idea is based off of Planet Labs' Flock Constellation of Dove 3U CubeSats that were used to produce high-resolution images of Earth. The problem lies with needing ground relay for the CubeSat constellation to produce accurate imaging. For Earth, ground relay is okay, but if this idea is to be scaled up to be applicable to any celestial body, ground relay needs to be eliminated. This requires good communication and cross-referencing between CubeSats within the constellation. This can be done by using data associated with the each satellite themselves. Error satellite attitude can be gauged by comparing feature points between images taken from different satellites and their estimates of attitude associated with a given image.

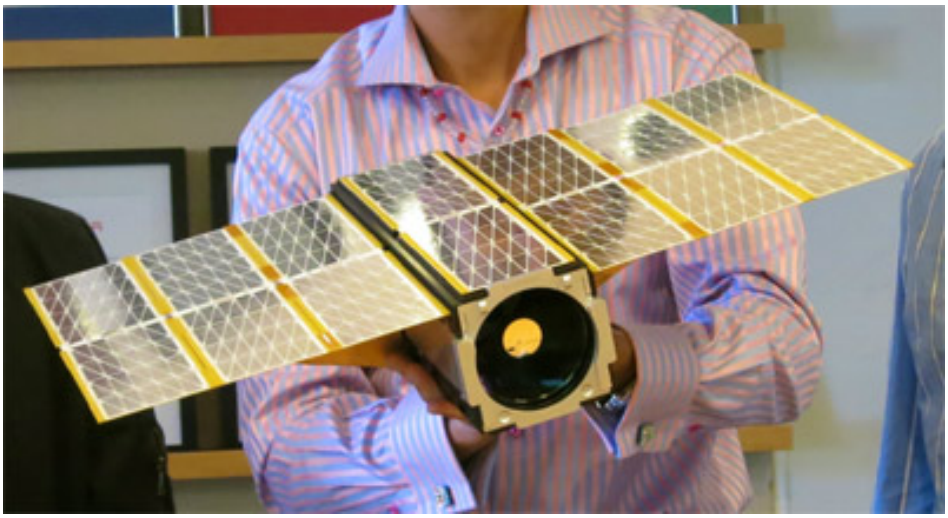


Figure 1: Dove CubeSat scale.

3 Methodology

3.1 Satellite Parameter Estimation

3.1.1 Dovesat Subsystems

Below are the various subsystems that will be found within the CubeSats in the constellation and what the components of those subsystems entail. The layout of subsystems is derived from the Dove 1 CubeSat design (Cosmogia Inc.):

1. Command and Data Handling will need at least one computer to perform computation. As ground relay will not be an option, hardware capable of more intensive computation will be required for the CubeSat which may mean a larger and heavier computer.
2. The Power Subsystem will be assumed to be similar to that of the Dove-1 cubesat in which it will possess 8 Lithium Ion cells that have 20Ah combined charge full capacity. The Power Subsystem will also have deployable solar cells for battery recharge.
3. The Attitude Control (ACS) Subsystem will use magnetometers, sun sensors, star trackers, horizon sensors as detection mechanisms and gyroscopes and magnetorquers for attitude control.
4. The Communications Subsystem consists of a very high frequency (VHF) radio for transmitting telemetry and an S-band frequency hopping spread spectrum modem for two-way communication and data downloading.
5. The Imaging system also needs to be taken into account which is largely just the camera.

3.1.2 Dovesat Mass Estimation

Below is the estimation of the mass of componenets in each subsystem and estimating a mass for the entire cubesat:

1. The mass of the shell of a 1U cubesat is 1.33 kg and linearly scales so a 3U cubesat shelling can be estimated to be at roughly **4 kg** (SMAD). This is assuming that the material for the shell of the cubesat is aluminum alloy 6061.
2. Unless a heavy high-powered computer is used on the cubesat to be able to keep up with the computation, range of masses for on-board computers averages out to about **100 g**.
3. The lithium-ion batteries average from searches to be around 60 g so for 8 batteries should be around **0.5 kg**.
4. There will be 7 deployable solar cells on cubesat and be 30cm x 10cm in surface area. The mass for a 3U solar panel is 155 g with a magnetorquer. The total mass is **1.085 kg** (EnduroSat).
5. Magnetometers, Sun Sensors, Horizon sensors, and the VHF radio are very light at less than 50 g. Assume all these components total to around **150 g**.
6. Assuming medium-sized CubeSats gyroscopes of 130 g are used and that there are 4 aboard with one as a redundant one, this total mass is **0.52 kg**.
7. Star trackers are a little heavier at around **250 g**. It is assumed that each cubesat is equipped with one.
8. The S-Band Transmitters are around **75 g**.
9. Cameras and imaging systems range from 0.5 kg to 1.2 kg the average will be taken at **0.85 kg**.

Summing all of these masses listed above, the total estimated mass of a CubeSat in the constellation is around **7.53 kg**. This is about 30 percent larger than the mass of the Dove CubeSat estimated

mass at 5.8 kg (Eoportal). This may be due to overestimates of the mass of CubeSat components. This may also be due to the need for heavier components such as computers because of the more computationally heavy problem solve of not having ground relay.

3.1.3 Moment of Inertia

Calculating the moment of inertia of the system was dealt with by using parallel axis theorem and rotation matrices. There were a few of assumptions made first before calculating the moment of inertia:

1. The moment of inertia with respect to the body frame does not change over time.
2. The moment of inertia is being calculated for the configuration when the solar panels are fully deployed in a "wing" formation as can be shown in Figure 1.
3. The mass distribution of the 3U CubeSat body with combined components and frame (not including gyroscopes) is evenly distributed or the density is constant with respect to spatial variation.
4. The moment of inertia will be calculated with these main parts of the assembly: the 3U CubeSat body without gyroscopes, the gyroscopes, and the solar panels.

With these assumptions, this is the process that was used to calculate the moment of inertia matrix of the CubeSat:

1. First the center of mass with respect to all relevent components is calculated in x,y,z coordinates in the body-fixed frame, β.
2. The centers of mass of the individual relevant components are identified.
3. Moment of inertia matrices with respect to β are found for all relevant components except for the gyroscopes.
4. For the gyroscopes, the inertia matrix is calculated with respect to the gyroscope's body fixed frame, A. After finding the inertia matrix with respect to A, the rotation matrix, $R_{A\beta}$,is found between A and β. Using the rotation matrix and its inverse, $R_{\beta A}$, the inertia matrix of each gyroscope can be converted to be with respect to the body fixed frame β as follows: $I_\beta = R_{\beta A} I_A R_{A\beta}$
5. Then parallel axis theorem can be used on each component with respect to the center of mass of the system. Parallel axis theorem is as follows: $I_i = I_{com,i} + md^2$ where d is the distance of the center of mass of component i from the total system center of mass.
6. The combined inertia matrix with parallel axis theorem for each component will be summed to represent the inertia matrix for the entire system.

Here below are the moment of inertia matrices of the 3U CubeSat body and solar panels with respect to the body frame. The parameters are:

1. The height of the CubeSat, h = 30 cm.
2. The width of the CubeSat, L = 10 cm.

3. The thickness of the solar panels: t = 150 μm

These moment of inertia matrices are calculated with respect to the body fixed frame β:

$$[I]_{cube} = \frac{m_{cube}}{12}\begin{bmatrix}(h^2+L^2) & 0 & 0\\ 0 & (h^2+L^2) & 0\\ 0 & 0 & (L^2+L^2)\end{bmatrix} \tag{1}$$

$$[I]_{sp} = \frac{m_{sp}}{12}\begin{bmatrix}(h^2+t^2) & 0 & 0\\ 0 & (h^2+L^2) & 0\\ 0 & 0 & (L^2+t^2)\end{bmatrix} \tag{2}$$

And here is the gyrostat inertia matrix calculated with respect to the gyrostat's body fixed frame, A, where r is the radius and h is the height. The actual parameter values are not definite yet:

$$[I]_g = \frac{m_g}{12}\begin{bmatrix}(3r^2+h^2) & 0 & 0\\ 0 & (3r^2+h^2) & 0\\ 0 & 0 & 6r^2\end{bmatrix} \tag{3}$$

3.1.4 Ballistic Coefficient

The ballistic coefficient is not relevant for most satellite applications in the short term but may be relevant in the long term due to effects from the atmosphere. The ballistic coefficient was calculated using this equation (Kilic):

$$BC = \frac{m}{C_D A_{ref}} \tag{4}$$

Where m is the mass of the "projectile" or CubeSat estimated to be 7.53 kg, C_D is the drag coefficient estimated to be 2.3, and A_{ref} which is the reference area interpolated to be 0.017505 m^2. Using these values the ballistic coefficient for a 3U CubeSat (neglecting deployed solar panels) is estimated to be **187.03** $\frac{kg}{m^2}$.

3.2 ACS Subsystem

The Attitude Control system mainly consists of magnetorquers and reaction wheels. Given both are working in tandem to adjust the attitude of the CubeSat, these two equations are the governing equations of the system:

$$I_{tot}\dot{\omega} + \omega \times (I_{tot}\omega + h_w) = -\tau_w + T_m \tag{5}$$

And:

$$\dot{h}_w = \tau_w \tag{6}$$

Where:

1. I_{tot} is the total moment of inertia matrix of the system.
2. ω is the rotational speed of the body fixed frame, β, with respect to the inertial frame, F.
3. h_w is the total angular momentum of the reaction wheels.
4. τ_w is the total torque produced by the gyrostats or reaction wheels.

5. T_m is the total torque produced by the magnetorquers.

Now assuming that the rotational speed of the body fixed frame with respect to the inertial frame does not move and that the reaction wheels are perfect and have no limitations so magnetorquers are not required in attitude calculations for the system, the governing equations simplify to this:

$$\omega \times (I_{tot}\omega + h_w) = -\tau_w \tag{7}$$

And:

$$\dot{h}_w = \tau_w \tag{8}$$

These two equations above will be used to calculate and simulate the attitude adjustment of a CubeSat with respect to its body fixed frame, β.

3.2.1 Magnetorquers

To be consistent with the full ACS the magnetorquers should still be included to have the most accurate prediction of the system, although the magnetorquers are not modeled in the ACS Matlab integration just yet. This is the governing generated torque equation for a magnetorquer:

$$T_m = -B \times M_d \tag{9}$$

Where B is the magnetic field vector with respect to the body fixed frame β and M_d is the magnetic dipole moment specific to the magnetorquer which can be assumed to be around 0.2 Am^2. Assuming imperfect gyroscopes and the presence of a magnetic field, the magnetorquers can be used to help adjust the attitude of the spacecraft.

3.2.2 Reaction Wheels

As already mentioned above, the governing ACS equation for a system when only gyrostats are present can be below, with a little bit of elaboration:

$$T_w = \omega \times (\omega I_{tot} + h_w) \tag{10}$$

Where T_w is the combined torque generated by all of the gyrostats or reaction wheels, ω is the rotational speed of the body fixed frame with respect to the inertial frame, I_{tot} is the total inertia matrix of the system with respect to the body fixed frame, and h_w is the total angular momentum of the gyroscopes. Total angular momentum of the gyroscopes can be summarized as:

$$h_w = \sum_{n=1}^{N} I_{w,n}^{W_n\beta}\omega_n^A \tag{11}$$

Where $I_{w,n}^{W_n}$ is the inertia matrix of the gyrostat in the gyrostat body fixed frame and ${}^{\beta}\omega_n^A$ is the rotational speed of the gyrostat n body fixed frame with respect to the satellite body fixed frame. Using these governing equations a matlab integration can be made to simulate the ACS system working to correct the attitude of a CubeSat.

3.2.3 Dynamics of Entire ACS with Matlab Integration

The matlab integration consists of these key elements:

1. Generating the appropriate inertia matrices for the total system and the individual gyrostats using functions.
2. Calculating the required torque needed to bring a system to zero rotational speed in the body fixed frame using functions.
3. Using an ODE solver to generate system state values at discrete time steps.
4. Developing a plot3 animated model of the cubesat that is able to rotated. Then use the outputed state values at discrete time steps to simulate the satellite coming to zero rotational speed in a designated time.

Obviously if this matlab integration were completely true to the actual CubeSat ACS then magnetorquers would be involved in the calculations. Magnetorquers should be integrated into the matlab but not until the gyrostat-only model has been mastered.

With this model there will also be inputs needed for the system. Here are the relevant inputs:

1. ${}^{I}\omega^{\beta}$: The rotational speed vector of the body fixed frame with respect to the inertial frame.
2. ${}^{\beta}\omega^{A}$: The rotational speed matrix of all the gyrostat body fixed frames with respect to the system body fixed frame.
3. A matrix of locations of all of the gyrostats' centers of mass.
4. A matrix of direction vectors of each gyrostat body fixed frame rotation axis with respect to the system body fixed frame, β.
5. Altitude of the CubeSat above the surface of the celestial body, in this case the test case is above Earth. This is used to calculate the magnetic field if magnetorquers are integrated.
6. The longitude of the CubeSat above the equator of the celestial body. This is used to calculate the magnetic field if magnetorquers are integrated.

3.3 ANSYS Modeling

The ANSYS modeling builds on work done by both Sarah Richter and Evan Wilt. The geometry was improved from a 1U cubesat to a 3U CubeSat with deployed solar panels and the Transient Thermal and Transient Structural simulations on the model were updated.

3.3.1 Designing Satellite in Spaceclaim

The SpaceClaim model of the CubeSat was updated to be 3U by extending the inner frame and outer plating from a 1U to a 3U design. Deployed solar panels were also included in the design to stay consistent with the most permenant configuration of the CubeSat. A view of the full assembly of 3U cubesat combined with solar panel can be seen below in Figure 2:

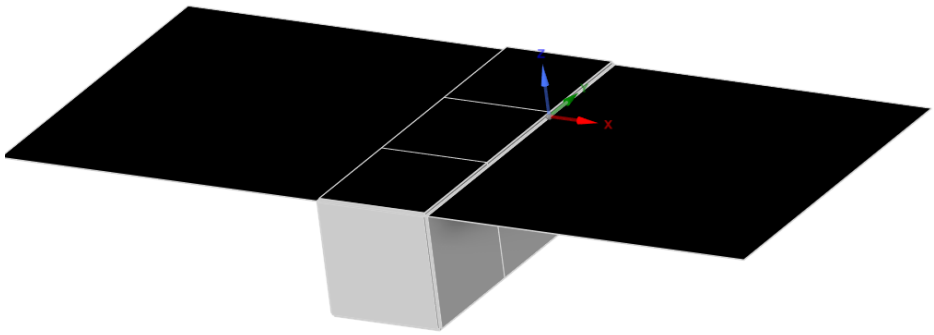
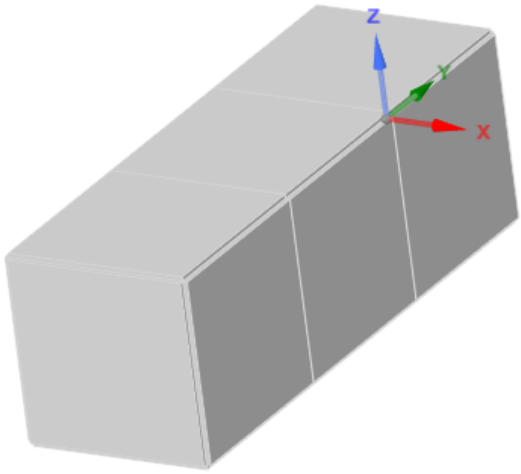
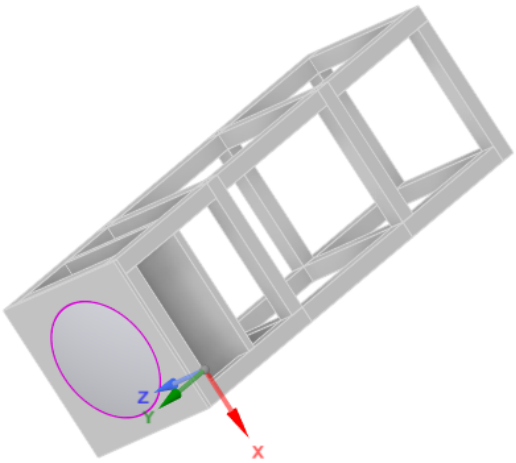


Figure 2: Whole CubeSat assembly.

And below in Figures 3a and 3b are different views of the CubeSat body without solar panels:



(a) CubeSat body with side plating.



(b) CubeSat body with only inner frame.

Figure 3

3.3.2 Material Estimation

Through Extensive research it was determined that the frame and plating of the 3U CubeSat body was made out of either Aluminum Alloy 7075 or Aluminum Alloy 6061. Evidence points towards the CubeSat frame being made of the latter so the CubeSat assembly components apart from the lens and solar panels were assigned to Al 6061.

The material for the imaging lens on the design was determined to be a silicate glass. The corresponding properties have not yet been assigned to the lens due to not knowing exactly what type of silicate glass the lens is.

Considering solar panels are complex in that they are made of multiple different materials, it is hard to assign one material to the solar panels to be used in the simulations. Evidence points towards the solar panels being majority silicon more research on materials needs to be done. Also a more intricate CAD model of the solar panels needs to be generated if the solar panels are to be assigned more than one material type. Because of this ambiguity of material type for the solar panels, the solar panels were suppressed for physics in the transient thermal and structural simulations.

3.3.3 Updating Transient Simulations

The portion of the transient simulations that needs to be updated is the Transient Thermal heat fluxes on certain faces. Since the heat fluxes are dependent on area and the CubeSat geometry has been updated to the proper 3U model, some of the heat fluxes on the outer faces of the CubeSat need to be updated. Below in Figure 4 is the incident heat fluxes or flows and radiation assigned to each of the six faces of the CubeSat:

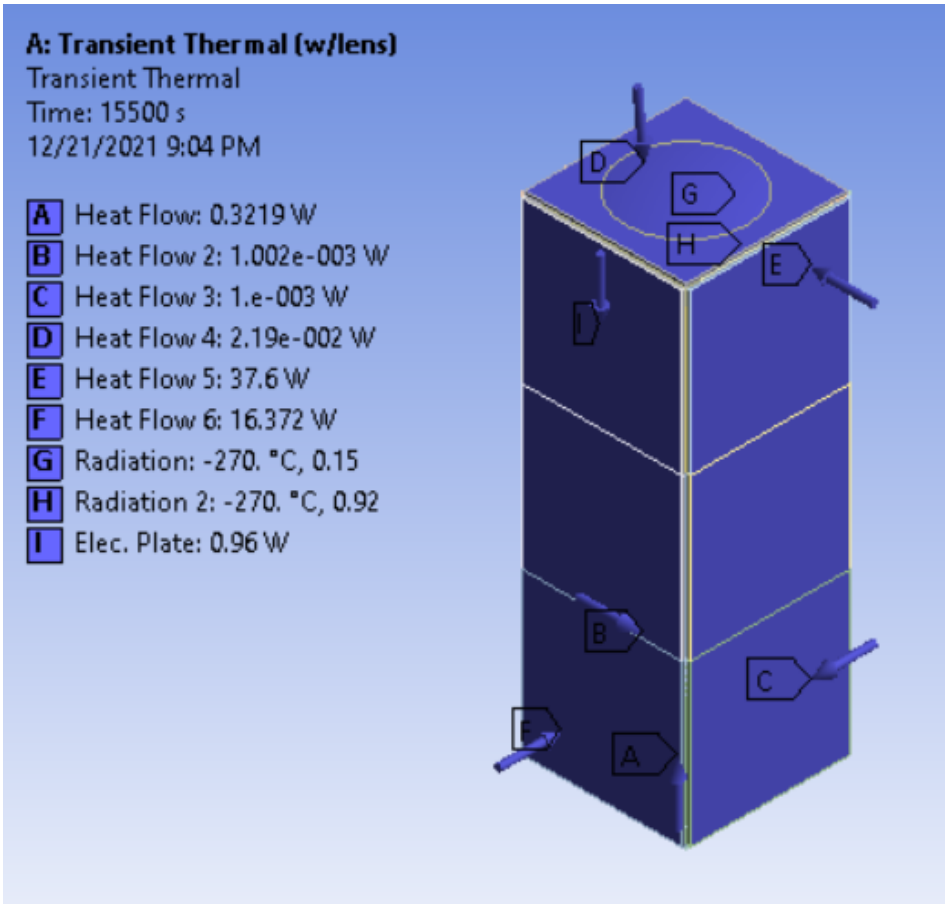


Figure 4: Transient Thermal heat flows and radiation assignments.

As can be seen above, items B, C, E, and F have all been updated by multiplying their corresponding values in the 1U CubeSat geometry model by 3. The Transient Structural simulation was not tweaked but needed to be re-ran with the new data points generated by the Transient Thermal solution over the 15500 second time span.

3.4 Filtering Simulation

43 unique images from 200 images were gathered and a time series of cropped images was extracted from each. The position of the cropped image was changed based on the dynamics of the satellite. The simulation does not consider the orientation of the satellite assuming nadir pointing, making the state the 3 Earth-centered Inertial positions and their ECI velocities. The cropped images are then shifted by the difference in longitude-latitude from the previous state. The feature points are extracted from each cropped image. The measurement model uses the (Longitude,Latitude,Altitude) LLA of these feature to estimate the movement of the image frame from the previous. This estimate is compared with the actual frames movement to generate the statistics. This estimate is than used as the measurement for the UKF.

3.5 Measurement Model

The first step in the measurement model is to extract features from the time series of images. There are 3 options for the feature detection SIFT (Scale Invariant Feature Transform), SURF(Speeded Up Robust Features), and ORB (Oriented FAST and Rotated BRIEF). SIFT was the original feature detector. Feature detector algorithms typically have four steps: determining scale space extrema, key point localization, orientation determination, and descriptor creation. SIFT uses a Difference of Gaussians (DoG) to execute the first two steps where the max and min from the DoG are the key points or features. The orientation in SIFT is determined by the local gradient of the image while the descriptor is a histogram of the gradient in different regions around the feature. SURF has increased speed verses SIFT by using more efficient algorithms. The order of SURF's four steps are: box filter not DoG, Hessian Blob-detection not DoG, gaussian wavelets not gradient, sum of gaussian wavelets not gradient histogram. SURF does typically detect less features than SIFT but still manages to do so robustly. The last algorithm is ORB. ORB was developed as a free to use alternative to SIFT and SURF. ORB uses FAST for feature detection and a multi-scale pyramid to provide semi-scale invariant. ORB then uses BRIEF to make descriptors and an intensity centroid to determine orientation.

After all features have been detected they are matched across images. The minimum unique matching algorithm (MUM) was made rather than using MATLAB's built in method because the MATLAB algorithm struggled to keep points between matches , low unique percentage, and improperly matched points at the edge of the image and features that looked similar. MUM was chosen rather than similar algorithms like Gated Nearest Neighbor and Gale-Shapely because it tends to not match outliers in the set more often. The time complexity of MUM is $n*m$ where n is the number of features previously and m is the number of current features. Gale-Shapley would have a time complexity of $max(n^2, m^2)$ and nearest neighbors a prediction time complexity of $max(n * log(m), m * log(n))$.

MUM first formulates a matrix of pairwise 2-norm distances between points. The i'th column refers to the i'th current point and the j'th refers to the j'th point. Two solutions on this matrix are then compared. The first is a minimization along the rows, matching previous points to their best

current point, the other does the opposite minimizing along the columns. For both solutions duplicates are removed so that each point has a one-one relationship or unique pair. A pair is unique if it does not share any index with another pair. Thus the maximum number of unique points is the minimum of the number of n and m. The solution with the least number of unique matches or smallest cumulative distance is selected. The locations of these matched points in the image are then subtracted and averaged. This returns the average movement of each point which is analogous to the movement of the frame as a whole.

3.6 Measurement Function

The measurement function is simply the difference in latitude and longitude between images. This is scaled by focal length in order for the measurement to be in pixels, like the measurement model, rather than kilometers, like the state. When implemented in the filter the measurement function converts the current and previous state from ECI to LLA and calculates the distance between them. Statistics for measurement model and function were gathered by subtracting the measurement model and measurement functions output from each other. The difference was then used to find the 4 statistical moments of the estimator using matlab built in functions for all 43 unique images.

3.7 Unscented Kalman Filter (UKF)

The UKF was chosen was because it posses the ability to deal with non-linear measurements and dynamics. This important because satellite dynamics are non-linear. Our measurement function is also non-linear because even though it is a linear geometric operation on the state, converting between frames.This is the measurement compares two different non-linear states at different times. It does however assume gaussian noise which is not something that guaranteed especially for the measurement function. The UKF accounts for non-linearity by having multiple Sigma points around the current estimate. It then propagates and measures these sigma points to form estimates of the state and measurement. The UKF then uses weight matrices and these estimates to calculate the state covariance , the measurement covariance, and the cross covariance. It then uses these covariances and the innovation, difference in measurement and measurement estimate, to update its state and covariance estimates. This is a similar but more complex process to the Kalman filter. The UKF was also gated such that measurement estimates that lay outside the chi squared inverse gate of 90% were not used.

4 Results and Discussion

4.1 Comparing Feature Detectors

The three figures below illustrate the results of using each feature detctor on the same image its distortion and and matching points between them. MATLABs matching algorithm was used. This algorithm uses the feature descriptor to match between points. For SIFT the algorithm identified 790 features in the first image and 939 in the second. Of those 790 182 were matched at unique percentage of 23%. For SURF 31 features in the first image and 29 in the second identified. Of those 31% were matched. For ORB the algorithm identified 940 features in the first image and

662 in the second. Of those 790 22% were matched. From visual inspection it is easy to see that in SURF and SIFT there were several erroneous matches were made. ORB seems to match more consistently than SIFT because the lines seem to more consistently go in the same direction near each other. ORB was the fastest of the 3 taking only 10 ms to run the feature extraction and matching. SURF took 33 ms and SIFT took 132 ms to run.

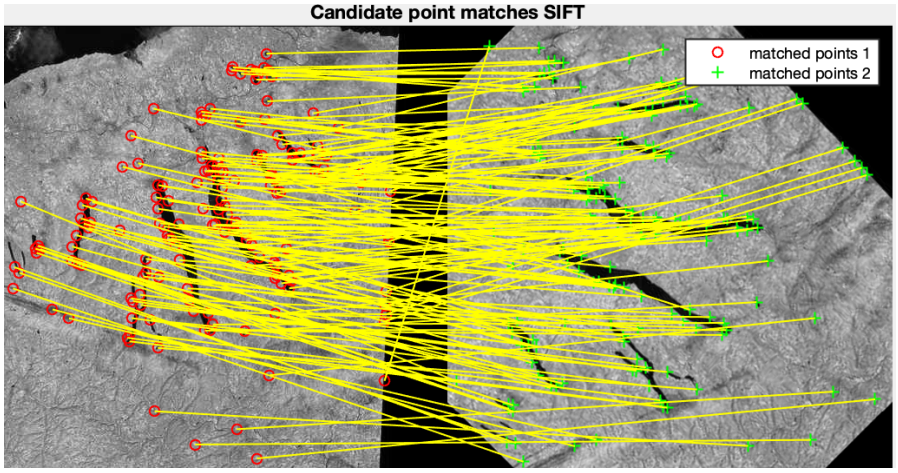


Figure 5: SIFT matching

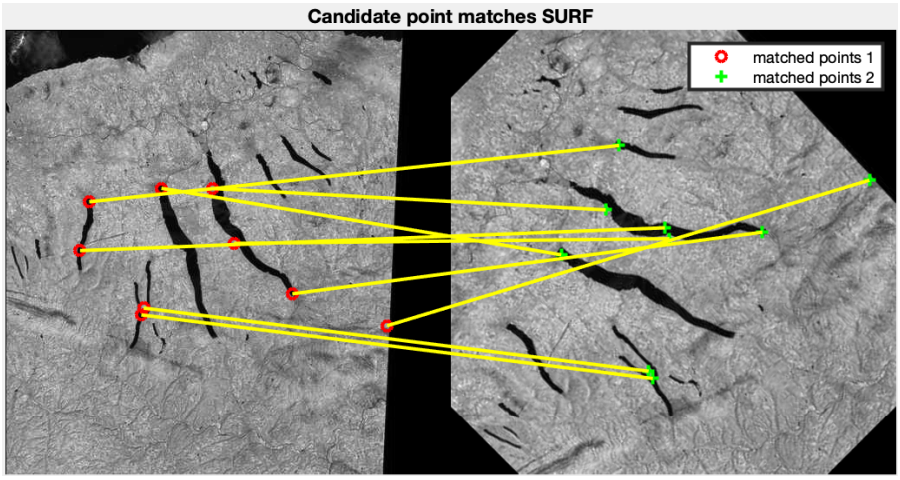


Figure 6: SURF matching

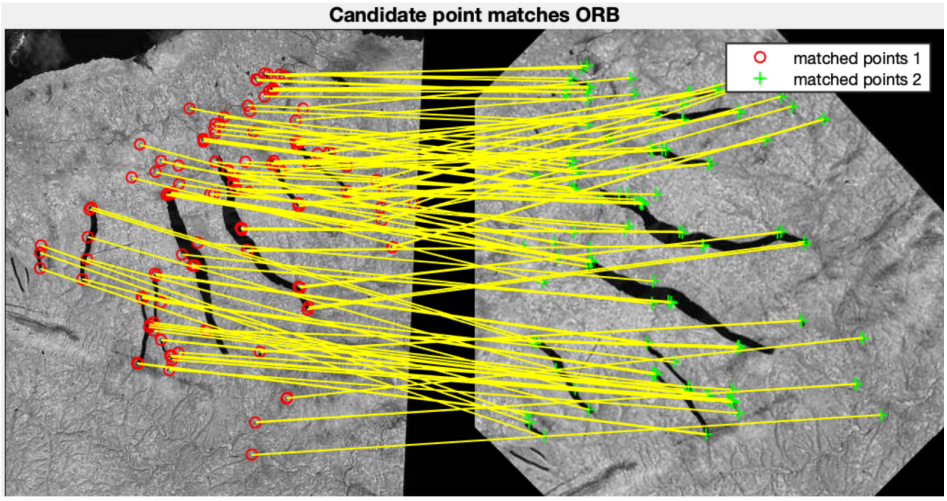


Figure 7: ORB matching

Figure 8 compares the resulting statistics for ORB and SIFT at an image size 150 pixels. Both had sub-pixel biases. SURF was not analyzed because for more than half of the images no features were detected. SIFT also had several images where no features were detected. There were some images where one or the other algorithm did not calculate statistics similar to other images in the set. For most statistics the two algorithms find similar results. The x-statistics for both were about $[0.004, 0.05, -4, 20]$. In y the statistics for ORB were typically $[-0.1, 0.18, -0.7, 2]$ while for SIFT they were $[-.0.01, 0.2, -.8, 2]$. The main difference here is that y-bias is much better for SIFT. The most jaring statistic would have to be the kurtosis. The kurtosis of a normal distribution is 3. A kurtosis of 20 in the x-direction means that the tails of the distribution are very heavy. The tails are probably this heavy do to rounding. The typical x measurement estimate is .05 but the frame only moves after a .5 pixel shift. This means that every tenth measurement is a 1 leading to heavy tails. This is as opposed to the y measurement estimate of .7. Rounding makes the measurement multi modal due to the sub-pixel movement.

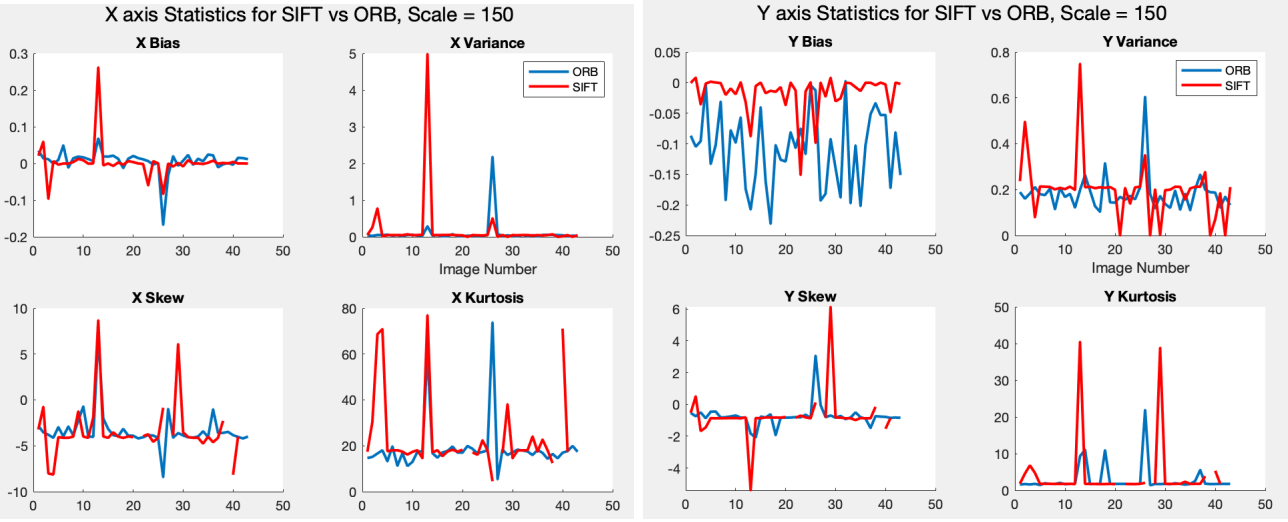


Figure 8: Measurement Statistics in Pixels

4.2 Varying Size

In figure 9 shows the result of testing ORB with images of differents sizes. Not much changes in the statistics only the y bias is different. Increasing scale tends to smooth out any outliers bringing all images to the same statistics. Increasing the scale tends to increase the y-bias and does not have the smoothing effect seen in other statistics. Further analyses would be needed at larger scales (the max being just under 2000) to see if the smoothing continued. This was not done as once the scale pased 200 runing the simulation takes an hour. The reason for this can be seen in figure 10. An increse in scale from 150 to 250 represented and order of magnitude increase in the number of detections. This explains the smoothing effect as there are more points to match. Also from figure 10 5 gaps can be observed in SIFT where no detections were made.

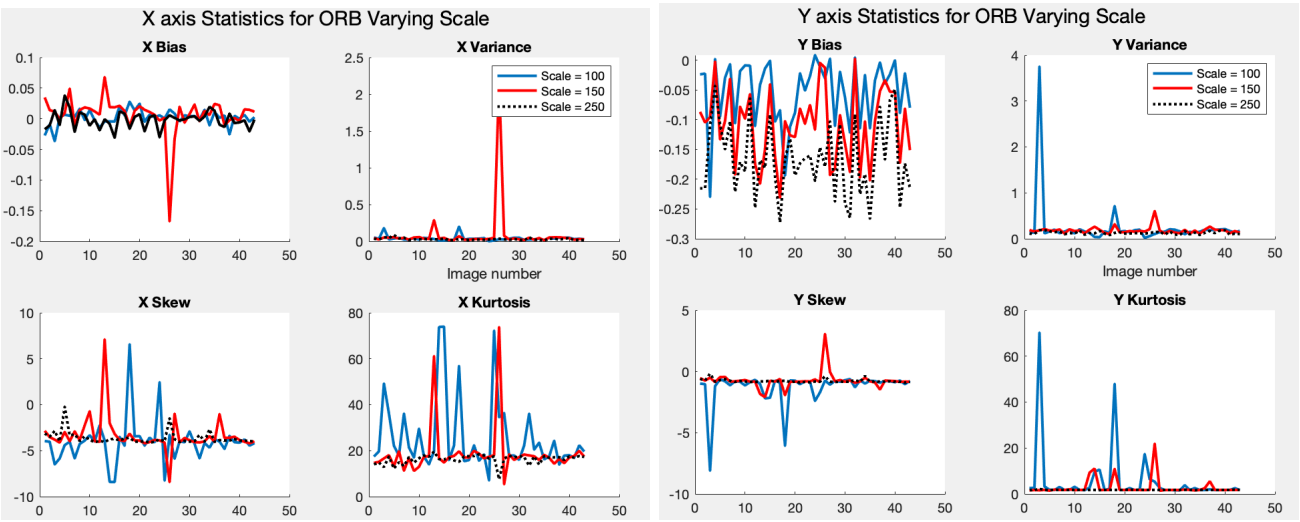


Figure 9: Measurement Statistics in Pixels for Varying Size

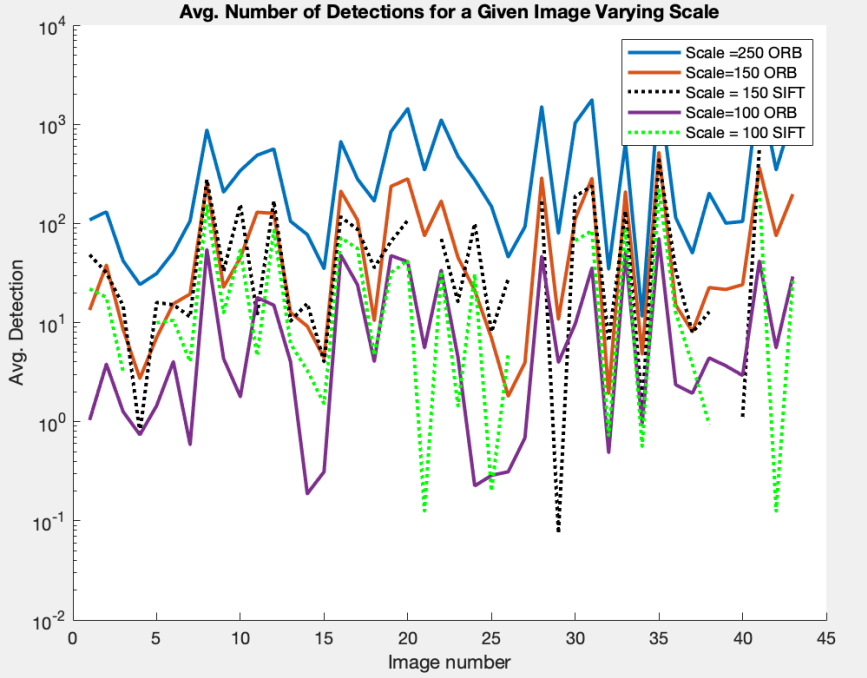


Figure 10: Detections at Varying Size

4.3 Perturbing State

Figure 11 illustrates how changing the magnitude of the velocity changes the statistics. Increasing the velocity changes the latitude and longitude more resulting in larger shifts. These larger shifts reduce the rounding problem discussed earlier and lead to movement on the scale of a pixel or more. In general reducing the velocity had the same smoothing effect as increasing the scale. Reducing velocity also reduced bias but tended to increase skewness and kurtosis. This helps explain the low kurtosis of the y measurement vs the x measurement since the y measurement at the original velocity is typically an order of magnitude greater than the x. Increasing the velocity also increased the amount of bias in the y quite significantly. This may not be that bad since because of the larger shift the percent bias may not change as much.

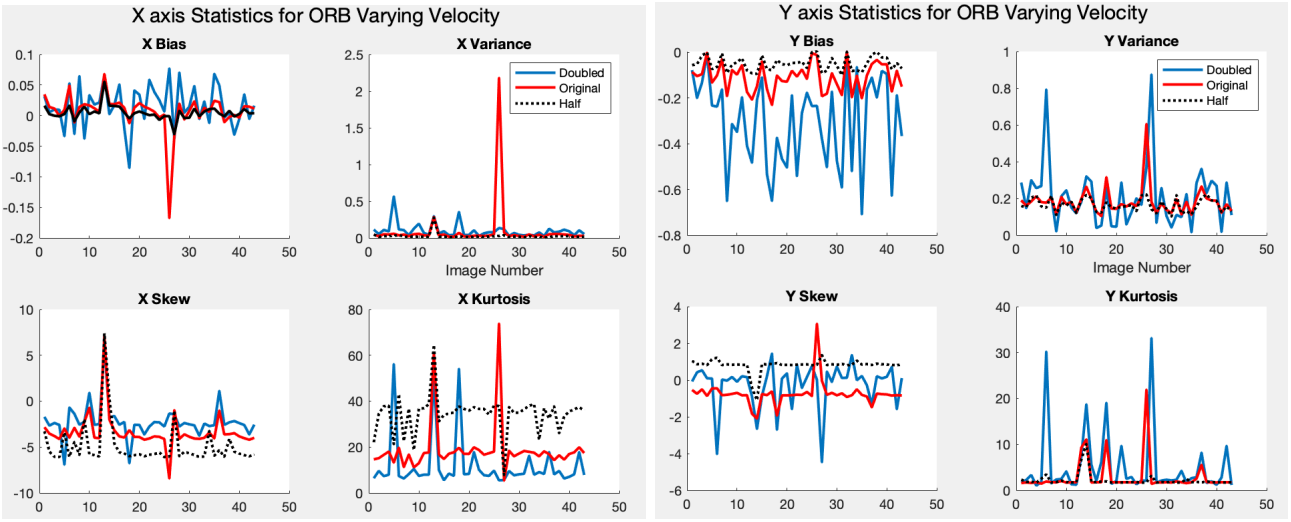


Figure 11: Measurement Statistics in Pixels for Varying Magnitude of Velocity

4.4 State Estimation

In general the error of the UKF's state estimate was on the order of 10 metetrs in position and meters per second in velocity. For the UKF Beta was set to 2 and Kappa to 0 as recommended while alpha was tuned. The alpha tuning in figure 13 was taken at a Q magnitude of E-7. Figure 12 was taken at an alpha of 0.5 at this alpha value none of the y biases were stable at this value. The E-9 value is the same process noise as Sam and Nathaniel's paper. In general in both figures 13 and 12 the filter was consistent with zero error always falling within the 2 sigma bound. The alpha =.25 and the smallest Q magnitude are barely consistent with the bound stratilling just the other side of zero as seen in figure 14. The larger alpha was the only one that was stable in all 6 errors as they all get smaller over time. Notably the error in x and z velocities never change remaining constant given the input parameters.

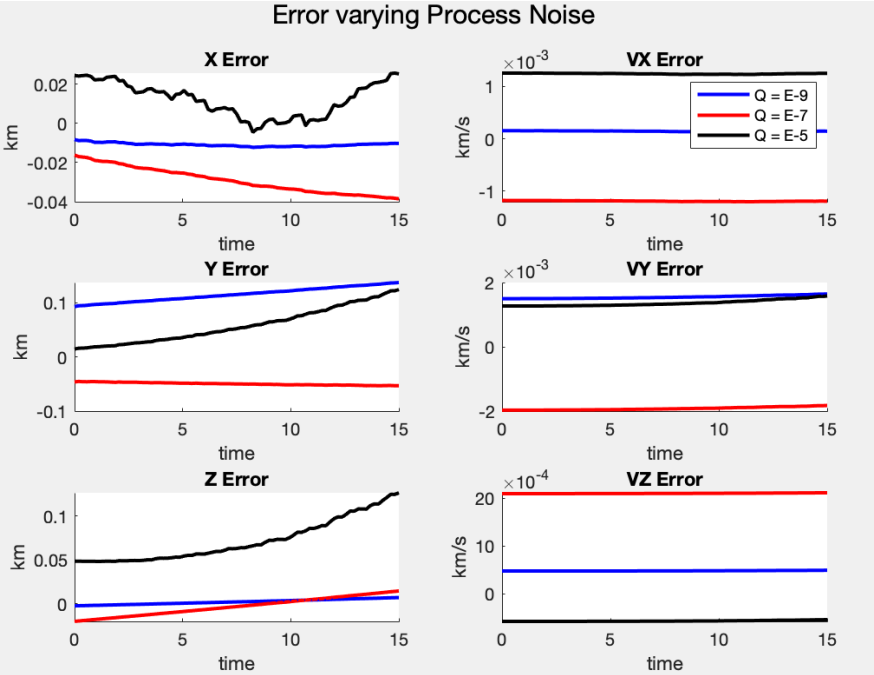


Figure 12: Error for Different Magnitudes of Process Noise

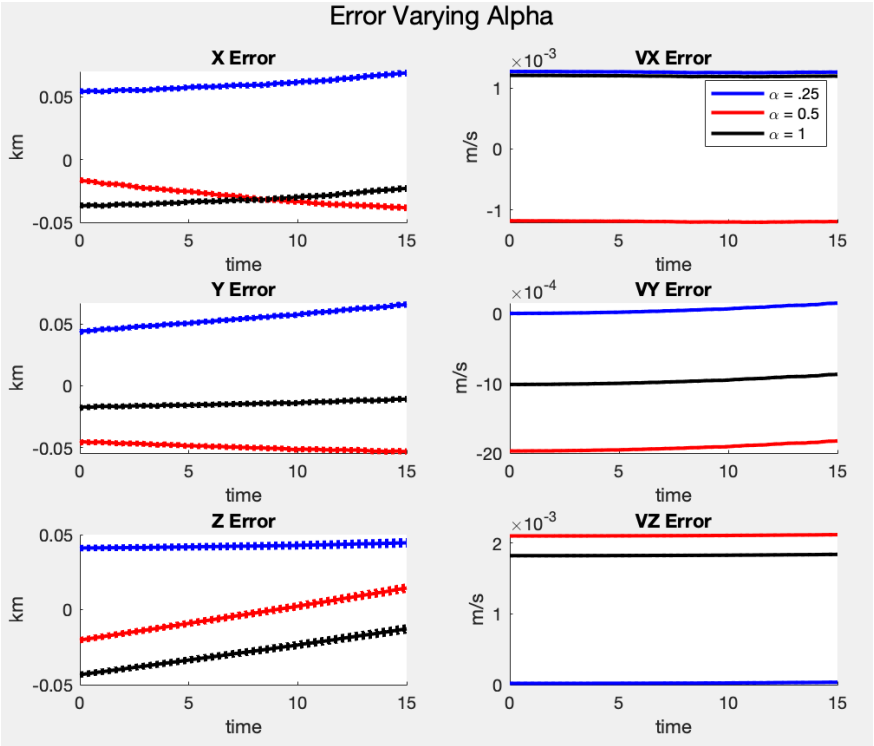


Figure 13: Error for different Alpha Tuning Parameters

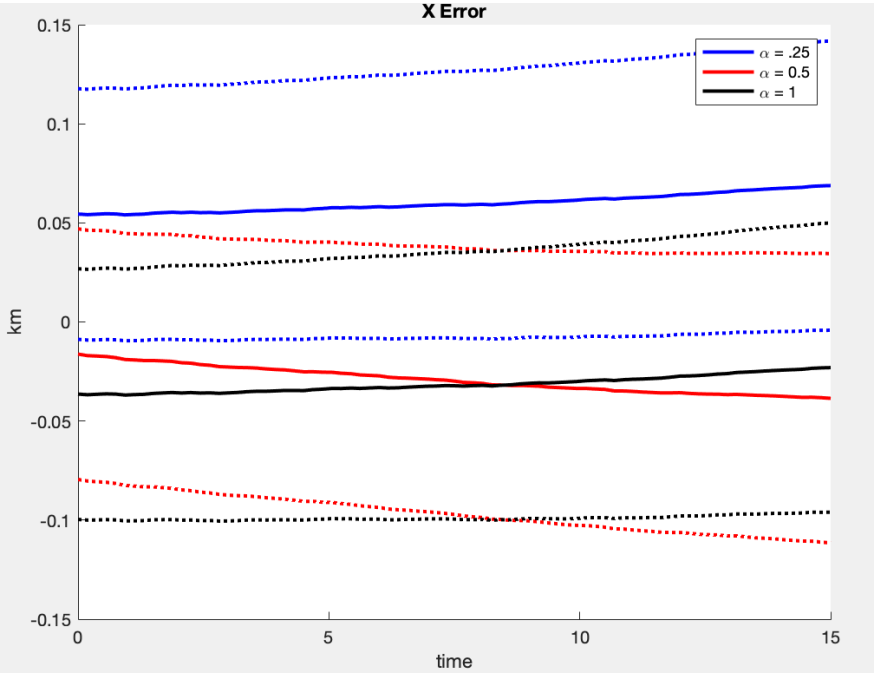


Figure 14: X Error for different Alpha

4.5 ANSYS Results Updates

Updating the geometry in SpaceClaim and updating the heat flow values in Transient Thermal, this is the final state (15500 seconds) transient thermal of the entire CubeSat body with the solar panels suppressed for physics in Figure 15:

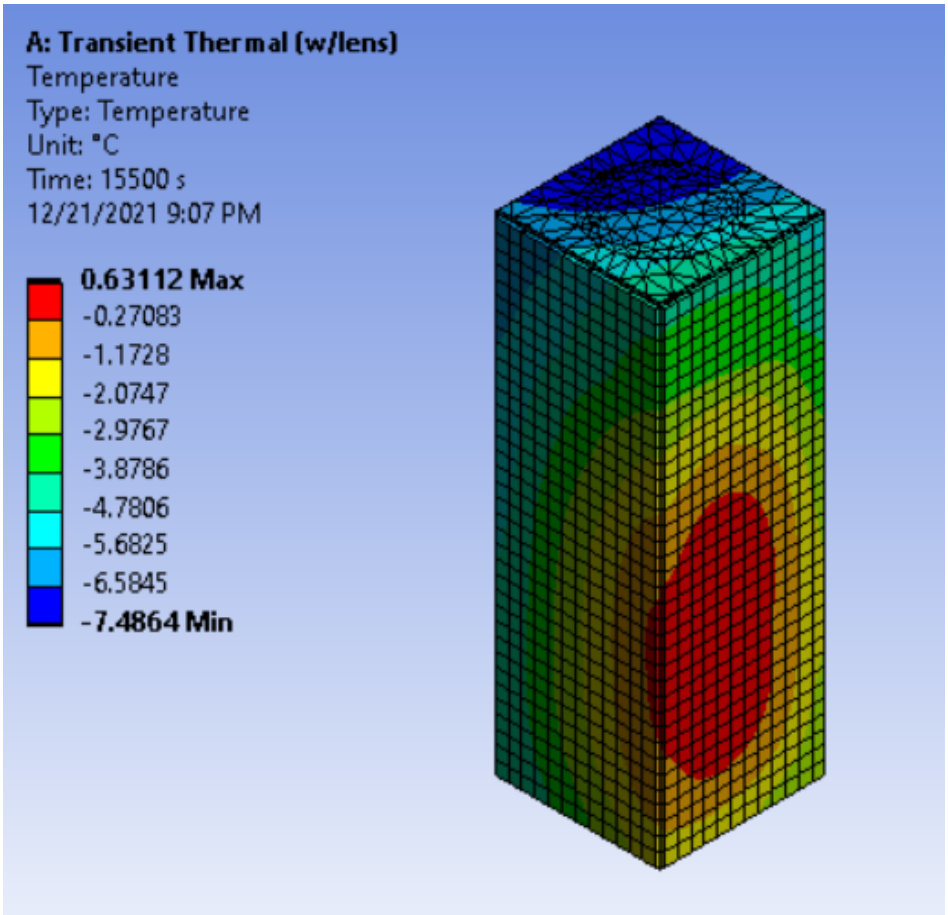


Figure 15: Final state total body thermal distribution

Comparing this to the data points collected for the 1U CubeSat geometry done prior the distribution looks different but this could be due to the geometry being large enough to where radiation effects dominate the thermal distribution. Below is a localized thermal distribution on the imaging lens of the CubeSat at the final state of 15500 seconds in Figure 16

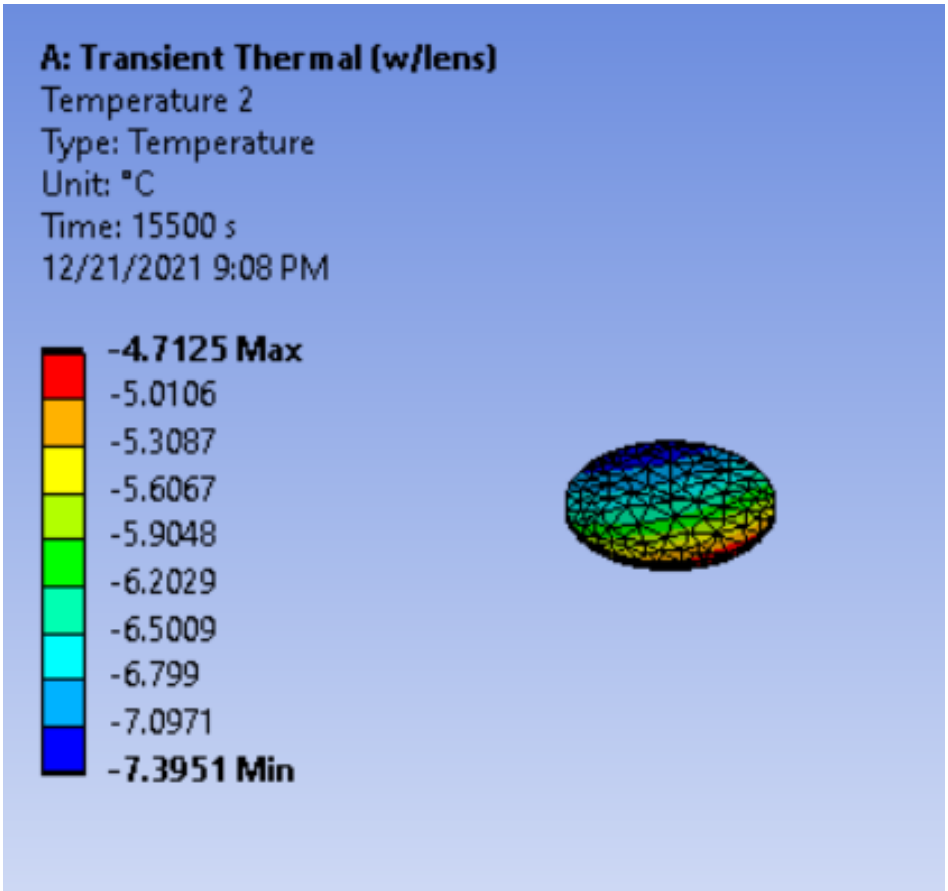


Figure 16: Final state lens thermal distribution.

The lens temperature distribution makes a little more sense.

Transferring the transient thermal solution to transient structural and then running the simulation, this is the final state transient structural deformation at 15500 seconds in Figure 17:

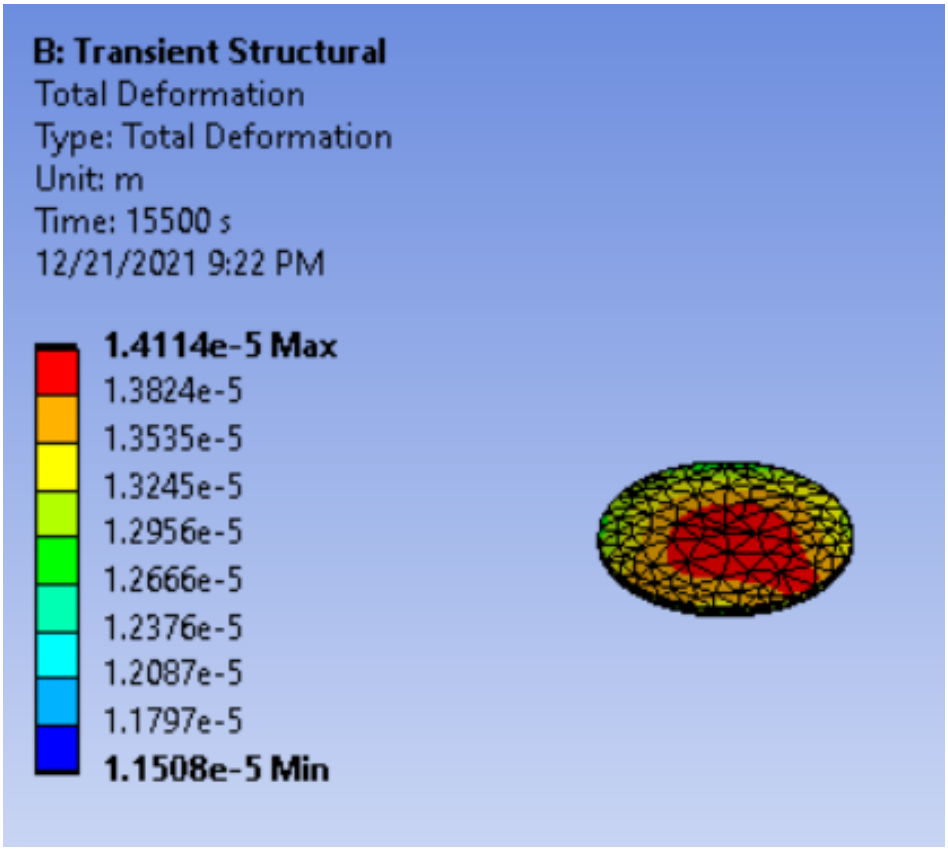


Figure 17: Final state lens deformation.

The deformation still aligns well with the simulation done on the 1U CubeSat. It is to be noted that the correct material type has not been assigned for the lens as of yet. In terms of next steps forward, the geometry could be better refined. This means more intricate parts and the addition of the assembly of parts within the 3U CubeSat body. With a more intricate geometry, this means a more intricate list of materials that will be assigned to the parts within the geometry assembly. The transient thermal and structural elements can be better refined as well in terms of heat flows and radiation emissivity for the transient thermal and better fixed support location for the transient structural.

4.6 ACS Matlab Integration Updates

The ACS matlab integration is not quite yet finished. This is what has been tentatively completed for the matlab integration:

1. Generating the appropriate inertia matrices for the total system and the individual gyrostats using functions.
2. Calculating the required torque needed to bring a system to zero rotational speed in the body fixed frame using functions.
3. Setting up ODE solver.

That leaves these items left to be finished:

1. Creating the function for the ODE solver to use.
2. Using the ODE solver to generate system state values at discrete time steps.
3. Developing a plot3 animated model of the cubesat that is able to rotated. Then use the outputed state values at discrete time steps to simulate the satellite coming to zero rotational speed in a designated time.
4. Integrate the option of having magnetorquers in the governing set of equations.

Although this will give a pretty accurate prediction of how the model will act, to have a fully accurate model other things need to be taken into account such as perturbations by other celestial bodies, finding the true mass distribution of the system, and micro-gravity. Simplifying assumptions can no longer be allowed if the true motion is to be simulated. Once the simplified matlab ACS integration has been mastered, there are many other steps needed to be at a fully representative model of the system. Especially when ground relay is not an option, accurate system state representation is crucial.

5 Conclusion

5.1 Parameters, ACS, and ANSYS

Estimating the parameters of the CubeSat was relatively straight forward as there were many references to help calculate them. The mass and spatial dimensions of the 3U cubesat body and solar panels was derived from the Dove CubeSats from the Planet Labs Flock Constellation. Although the mass of the system was overestimated by 30 percent, this estimation can be refined by having an exact parts list of a Dove CubeSat which would be an almost identical match to the CubeSats designed for imaging any celestial body. This desired parts list was not found in the discovery process. The only difference from a Dove CubeSat would possibly be the larger computer due to more complex calculations required because of no ground relay. The ballistic coefficient was pretty straight forward to calculate but could possibly be off due to interpolation.

The ACS was straightforward identifying the mechanisms that would adjust the attitude of the CubeSat, gyrostats and magnetorquers, but was hard to implement into a matlab master script that could simulate the attitude adjustment of the CubeSat. Using the governing equations in an ODE solver to generate state values of position and rotational speed for discrete time steps is not necessarily as easy as it seems. The next steps forward is completing the ODE solver and running an animation of the satellite coming to zero rotation speed in the body fixed frame. After that this model is mastered, the option for magnetorquers can be implemented and perturbations that can alter attitude over time can be implemented to fully describe the system realistically.

The ANSYS transient thermal and transient structural for the 3U CubeSat geometry was pretty straightforward to set up as the exact same simulations have already been done on the 1U CubeSat geometry. There are a few of things to do with this simulation moving forward. The first is to refine the geometry to where the parts of the assembly almost exactly resemble the parts of an actual

Dove CubeSat which will require a parts list. The assembly would need to include the parts of the CubeSat that lie within the 3U CubeSat body as well. Secondly with more intricate parts comes a larger list of materials that need to be assigned to parts. Especially when an assembly can have dozens of materials, this can be painstaking. Lastly, the heat fluxes or flows on the CubeSat can be refined when the solar panels are no longer suppressed for physics. This is because some parts of the CubeSat will no longer be illuminated as well with the solar panels shading them from sources of heat.

5.2 Filtering

The measurement function benefits from being a simple calculation of differences in the LLA frame while the measurement model has several improvements or ideas that can be tested to improve performance particularly of the skew and kurtosis. One idea for improvement is to use SURF as the feature detector on images at larger scales. The nature of the problem formulation necessitated smaller size images in order to shift them around the image plane. This SURF would have the advantage of having fewer features detected and are well separated from each other. This reduction in clutter is what allowed less bias in the y direction. As long as there are consistently detections of the order of 100 the smoothing effect should take place without a significant loss in separation. The increasing size will also allow for measurements to be on the order of multiple pixels. This should have the effect of lowering the skew and kurtosis as was the case for increasing velocity. If this does not work to make the distribution more gaussian then higher moment filters will be needed.

SURF does still have a potential problem with missed detections as it does not perform as well as SIFT which already had several missed detections. Using ORB, which performed the best of the three, at scale may be difficult due to high clutter. ORB at scale is also concerning since MUM and the other matching algorithms have poor time complexity. MUM with ORB at a size of 250 took an hour to run. To continue using ORB either a new matching algorithm must be implemented that is fast and robust even in high clutter environments or the number of points fed into MUM must be reduced. Removing points before processing must be done in some systematic way. When thresholding and only keeping the first 100 points or even a random set of 100 points the bias of the estimate oscillated wildly from 20 to even 60 pixels. Typically this would be done using the birth and death model of some tracking algorithm, this may not work for our implementation since we are explicitly attempting to compare only two images, the time series cropping served as an analogy to this. A solution for removing the number of points fed into MUM is to cluster points that are near each other in to a super feature or object. Another is to focus on only one region of the image at a time. Alternative algorithms to MUM are to implement an information filter or visual based SLAM and skip matching all together.

Other extensions of the measurement model should also be made. In order to relax the nadir pointing assumption the ECI to LLA conversion should be paired with a line of sight calculation. Adding this functionality will allow the estimate to take into account the satellite's pointing orientation. By using a star tracker or other measurement of the angular orientation the addition of line of sight can be used to estimate the orientation. This important because we want to be able to adjust pointing in order to capture the same image as another satellite. Another thing this simulation is missing is distortions to the image. This is important to test how robust the measurement model

and function are to radial distortion and blur that are inherent to this style of imaging.

6 References

Dove 1 Satellite Technical Description. Cosmogia Inc.

Dove - Satellite Missions - Eoportal Directory

"3U Solar Panel CubeSat Solar Panel." CubeSat by EnduroSat, 27 Oct. 2020

Kilic, Cagr. Deployment Strategy Study of QB50 Network of CubeSats

Wertz, James Richard, et al. Space Mission Engineering: The New Smad. Microcosm Press, 2011.

ORB: https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief

SURF: https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7

SIFT: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform