Satellite Image Filtering Masters of Engineering Report

Nathaniel Kinzly and Samuel Feibel

1 Abstract

Satellite state estimation is critical to the success of any space mission. In an imaging mission, a concern is the calibration of the onboard camera in order to take the clearest possible images. In order to solve these problems simultaneously and autonomously, we augment our spacecraft state to include information about both the dynamics of the satellite and the parameters of its camera. Through the use of an Unscented Kalman Filter (UKF), the images taken by the camera, the GPS, and attitude sensor measurements can be synthesized with the predicted dynamics of all states to produce estimates of both the dynamics and the current camera parameters. Two important aspects of this system are the orbit propagation model used to define the truth state and predict the satellite's motion and the measurement function used to predict the image observations from state data. We take two different approaches to creating a measurement function: a least squares approach and a neural network. We test the performance of each measurement function on a test set of data. We found that though both models gave nearly zero mean predictions, excluding specific data from the training data led to biased errors. Therefore, a more complicated model would likely be needed for beneficial UKF performance.

2 Table Of Contents

3 List Of Figures And Tables

List of Figures

4 List Of Symbols

$Y_{l,i}$	Neuron value i in layer l in a neural network
W_{ij}	Weight from $Y_{l-1,i}$ to $Y_{l,j}$ in a neural network
B_{ij}	Bias term from $Y_{l-1,i}$ to $Y_{l,j}$ in a neural network
$\mathbf{r}_{G/O}$	Radius vector from point O to point G
μ	Standard gravitational parameter of Earth
$\mathbf{u}_G$	Control force acting on point G
$\mathbf{f}_G$	External disturbance force acting on point G
$\mathbb{I}_{\mathbb{G}}$	Moment of Inertia about point G
${}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}$	Angular velocity of frame $\mathcal{B}$ in frame $\mathcal{I}$
$\mathbf{M}_G$	Control torque acting on point G
τ_G	External disturbance torque acting on point G
U	Non-spherical gravity field potential
$R_\oplus$	Radius of the earth
r	Orbital radius of satellite
P_l^m	Associated Legendre polynomials
C_l^m,S_l^m	Tabulated coefficients used in Earth's non-spherical gravity potential model
θ	Azimuth angle in spherical coordinates
ϕ	Zenith angle in spherical coordinates
$\mathbf{e}_n$	Unit vector normal to the satellite's plane of motion
$R_z(\theta)$	Right handed rotation matrix of an angle of θ about the z axis
D	Drag force magnitude
C_D	Coefficient of drag
V	Magnitude of the spacecraft's velocity with respect to the Earth's atmosphere
S	Spacecraft area
T	Temperature of Earth's atmosphere
$F_{10.7}$	Time varying property of the atmosphere
A_p	Time varying property of the atmosphere
m	Average Molecular Mass

h	Altitude
H	Correlated altitude in specific atmosphere
ρ	Density of the atmosphere
$\mathbf{r}$	Radius vector of the spacecraft's position
$\mathbf{v}$	Velocity vector of the spacecraft
${}^{\mathcal{B}}\mathbf{q}^{\mathcal{A}}$	Quaternion from frame $\mathcal{A}$ to frame $\mathcal{B}$
${}^{\mathcal{B}}C^{\mathcal{A}}$	Direction cosine matrix from frame $\mathcal{A}$ to frame $\mathcal{B}$
X_p, Y_p	x and y pixel location within an image
S_x, S_y	Width and height of an image in pixels
$B_{\theta 1}, B_{\theta 2}$	Left and right longitude bounds of an image
$B_{\phi 1}, B_{\phi 2}$	Top and bottom latitude bounds of an image
f	Camera focal length
c_i	Spherical camera distortion parameters
$\mathcal{X}$	Set of inputs to a measurement function
$\mathcal{Y}$	Set of outputs from a measurement function
$\mathbf{\Theta}$	Unknown parameter matrix in least squares model

5 Acknowledgements

We would like to acknowledge Professor Dmitry Savransky for his support and guidance on this project. We would also like to acknowledge Zvonimir Stojanovski for his work in support of this project.

6 Introduction

CubeSats and CubeSat constellations have become increasingly popular in the aerospace industry. In order to model the satellite's operations, its dynamic state is defined as a vector with its position, velocity, orientation, and angular rate. Similarly, for the satellite's camera, we designate a set of 4 parameters that define the camera's state. For effective operation, a satellite needs a strong estimate of both the dynamic and camera states. A filter can be used to synthesize known satellite dynamics with measurements made by sensors in order to generate estimates of all satellite states. Generally, this is done only for the dynamics states with measurements from GPS and attitude sensors. By augmenting the dynamics states with the camera states, the satellite could use the images taken of the ground in the measurement function in order to more accurately estimate both the camera parameters as well as the other satellite states like position and attitude.

We have generated two elements of this filter that are necessary for its operation and development. The first is the dynamics model that predicts the position and attitude of a low earth orbit (LEO) CubeSat. Variations of this model are used to define the true trajectory of the satellite as well as the time update prediction step in the filter itself. The second is a measurement model for the imaging observations that predicts observations of the camera given different position, attitude, and camera parameter states.

7 Literature Review

In the realm of nonlinear estimation, there are three major types of filters used: Extended Kalman Filters (EKF), Unscented Kalman Filters (UKF), and Particle Filters (PF). EKFs maintain simple update rules that make use of the original Kalman Filter (KF) equations. However, they rely on Jacobians of the dynamics and measurement which can lead to significant errors if the function is highly nonlinear [1]. In order to avoid errors that come from linearizing a function, Julier and Uhlmann developed the UKF which approximates the probability distribution at each time with a small set of sigma points [1]. The distribution of measurements and the distribution of states at the next time step are found by applying the nonlinear measurement and dynamics functions respectively to the sigma points. The UKF maintains similar computational complexity as the EKF while the filter can also be extended to include higher probability moments like skew and kurtosis [2]. PFs are a Monte-carlo method that approximates the state distribution by a random set of particles [3]. The computational complexity of PFs increases with the size of the particle set, so there is a trade off between computational time and accuracy of the distribution approximation.

There has been work done in the intersection of dynamic filtering and feature extraction by Monica and Nigel [4] and by Velastin and Xu [5]. The former work makes use of a background subtraction algorithm in combination with a KF to track a moving object in a sequence of video frames. The latter work uses an EKF in combination with a Hough transform for feature extraction to create a more robust feature extraction algorithm.

8 Methodology

8.1 Unscented Kalman Filter (UKF)

Due to the fact that the UKF can be augmented to account for higher order probability moments, the UKF was chosen to be explored for this system.

The UKF itself uses the nonlinear measurement and propagation functions of the system in order to develop its estimate. Here, our state consisted of both dynamic states (position, velocity, attitude) as well as camera states. In order to use the UKF, measurement functions for both the dynamic states and the camera states had to be generated to convert the current state to the expected measurement. One assumption of the UKF is that the measurement function has a Gaussian error distribution with zero mean error, which is an important constraint for our measurement function. For the dynamic states, the position, velocity and attitude were directly measured. For the imaging states, the measurements were feature-sets extracted from images of the ground. The feature-sets were generated using a scale invariant feature transform algorithm.

8.2 Scale Invariant Feature Transform (SIFT)

The SIFT algorithm extracts features, known as keypoints, from an image. These features are located at local maxima or minima of the image. A keypoint is uniquely identified by a descriptor, which is a 128 dimensional unit norm vector that contains information about the gradient around the keypoint. Each keypoint also has a scale, orientation, and sub-pixel position. As the name of SIFT suggests, the keypoints it finds are independent of translation, rotation, and scale of the image. The distance between two keypoints is equal to the Euclidean distance between their descriptors, and the dot product between two descriptors measures their similarity. Therefore, we can test to see if the same keypoint exists in multiple images by running SIFT on both images and examining the similarity between the two sets of keypoints to look for matches. This functionality allows for the development of a measurement model that relates features in different distorted version of the same original image.

8.3 Measurement Function

In order to predict the features from the satellite's state, we first had to isolate a single, primary feature, from all available features. The process for doing so is elaborated in section 9.2.1. Using this one to one relation between states and measurements, a general function was created to intelligently convert any state to a predicted measurement. Given that this function was highly nonlinear and difficult to define, two general methods were pursued to generate this function. Both methods used a training set to tune the function itself. These approaches were batch least squares and a neural network.

8.3.1 Neural Network

Neural networks are a way of finding a functional mapping between an input set $\mathcal{X}$ and an output set $\mathcal{Y}$ through the use of a large number of learnable parameters. Neural networks contain several different layers each with some number of neurons. In a fully connected layer, the value at each neuron is a linear function of the values at the neurons of the previous layer. This relationship is given in equation 1

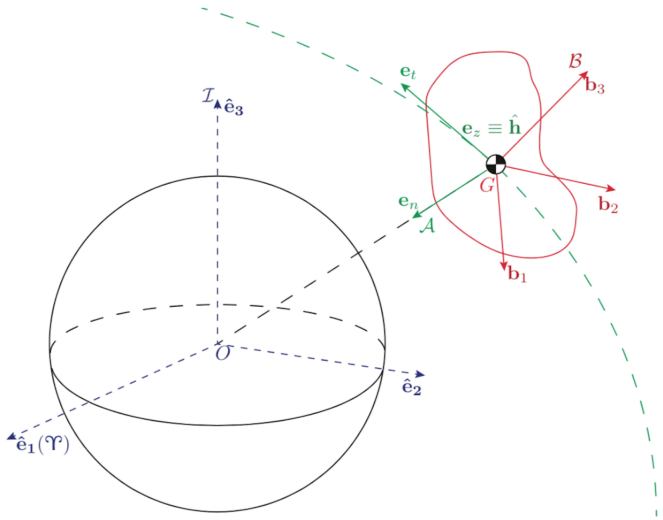

Figure 1: A diagram of the spacecraft Body frame (red), global frame (blue), and orbit local frame (green) for satellite dynamics [7]

$$Y_{l,i} = \sum_j W_{ij} Y_{l-1,j} + B_{ij} \tag{1}$$

where $Y_{l,i}$ is neuron value i in layer l, W_{ij} is the weight from $Y_{l-1,j}$ to $Y_{l,i}$ and B_{ij} is the bias term between these two neurons.

Training the neural network consists of two steps: forward propagation and backward propagation. In forward propagation, the output of the neural network is computed for some input and the current set of weights and biases. By comparing the output of the network with the true output, the network calculates a loss function. Then, in backward propagation, the network computes the gradient of the loss function with respect to all the weights and bias terms and uses that result to update the weights and biases as to minimize the loss function.

8.4 Satellite Orbit Dynamics

The filtering system requires a model of the spacecraft's dynamics for two reasons. The first is to use as the dynamics update in the time update step of the filter itself. The second is to use as the truth model in order to simulate the filter.

The simulated spacecraft travels in LEO with a low inclination, a similar orbit to that of the International Space Station. The dynamics of a rotating satellite in orbit follow equations 2 and 3 with geometry shown in figure 1

$$^{\mathcal{I}}\frac{d^2}{dt^2}\mathbf{r}_{G/O} + \frac{\mu}{||\mathbf{r}_{G/O}||^3}\mathbf{r}_{G/O} = \mathbf{u}_G + \mathbf{f}_G \tag{2}$$

$$^{\mathcal{G}}\frac{d}{dt}(\mathbb{I}_{\mathbb{G}} \cdot {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}) = \mathbf{M}_G + \boldsymbol{\tau}_G \tag{3}$$

where in equation 2, $\mathbf{r}_{G/O}$ is the vector pointing from the center of the earth to the satellite, μ is the gravitational constant of earth, $\mathbf{u}_G$ is the force provided by the satellite's attitude determination and control system (ADCS) and $\mathbf{f}_G$ is the external force on the satellite. In equation 3, $\mathbb{I}_\mathbb{G}$ is the moment of inertia matrix of the spacecraft about its center of mass, ${}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}$ is the angular velocity of the spacecraft, $\mathbf{M}_G$ is the torque provided by the spacecraft ADCS and $\boldsymbol{\tau}_G$ is the disturbance torques.

The most significant disturbance sources for a LEO satellite are Earth's non-spherical gravitational perturbations, atmospheric drag, and point-mass gravity from the moon and sun. Non-spherical gravity from earth is described by the potential in equation 4

$$U(r,\theta,\phi) = \sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{R_\oplus}{r}\right)^l P_l^m(\cos\phi)\times\left(C_l^m\cos(m\theta)+S_l^m\sin(m\theta)\right) \tag{4}$$

where $R_\oplus$ is the radius of earth, r, θ, and ϕ are the range, azimuth, and zenith of the spacecraft, P_l^m are the associated Legendre polynomials and C_l^m and S_l^m are coefficients of Earth's potential model.

The only significant torque disturbance was the 1st order torque of the gravity gradient from Earth described by equation 5.

$$\mathbf{M}_G^{(1)} = \mu\frac{3}{||\boldsymbol{r}_{G/O}||}\mathbf{e}_n\times\mathbb{I}_G\cdot\mathbf{e}_n \tag{5}$$

9 Experimental Details

9.1 Dynamics Model Implementation

The dynamics model propagates an initial satellite state of the form $\begin{bmatrix}\boldsymbol{r}_{G/O} & {}^{\mathcal{I}}\boldsymbol{v}_{G/O} & {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}} & {}^{\mathcal{B}}\mathbf{q}^{\mathcal{A}}\end{bmatrix}^T$ where ${}^{\mathcal{I}}\boldsymbol{v}_{G/O}$ is the velocity of the spacecraft in the inertial frame and ${}^{\mathcal{B}}\mathbf{q}^{\mathcal{A}}$ is the quaternion between the $\mathcal{B}$ and $\mathcal{A}$ coordinate frames. $\mathcal{B}$ is the body frame that travels with the spacecraft. $\mathcal{A}$ is the frame fixed at the spacecraft's origin with $\mathbf{e}_t$ pointing tangent to the orbit, $\mathbf{e}_r$ pointing anti-parallel to the radius vector, and $\mathbf{e}_z$ aligned with to the orbit's momentum vector. The initial time for propagation is given as a Julian date in order to specify the positions of the Moon and Sun relative to the Earth. A time vector for the range of time simulated is also specified.

These arguments are passed to the *ode113* [9] propagator in MATLAB which propagates the state over the specified time span using the natural dynamics and including all relevant perturbations.

9.1.1 Earth Gravity

The first perturbation calculated is Earth's non-spherical gravity. The general potential shown in equation 4 was approximated with coefficients up to 8th order [10]. The gravitational acceleration is the negative gradient of the potential. Due to the number of terms in the calculation, we chose to take advantage of MATLAB's symbolic toolbox. By formulating the potential in symbolic form and converting to Cartesian coordinates, the gradient could be taken symbolically and converted to a MATLAB function. This took a significant amount of time for MATLAB to execute but produced a function that could be executed quickly. The position also had to be converted to Earth centered Earth fixed (ECEF) from Earth Centered Inertial (ECI) when passed into this function because

the gravitational potential is inherent to the Earth's shape and therefore only applies in the Earth fixed frame. This conversion is shown in equation 6

$$\boldsymbol{r}^{ECEF} = R_z(\theta_{GMST})\boldsymbol{r}^{ECI} \tag{6}$$

where R_z represents the right handed rotation matrix about the z axis and θ_{GMST} is the angle between Greenwich Mean Meridian and the vernal equinox. The gravitational acceleration is then converted back to ECI for state propagation.

9.1.2 Atmospheric Drag

The next significant perturbation implemented in the model was drag. The drag force in equation 7 is dependent on the atmospheric density ρ.

$$D = \frac{1}{2}C_D\rho V^2 S \tag{7}$$

where C_D is the coefficient of drag, V is the magnitude of the spacecraft velocity with respect to the atmosphere, and S is the area of the spacecraft. The atmosphere itself behaves unpredictably but can be approximated with equations 8 [11].

$$\begin{aligned} T &= 900 + 2.5 \times (F_{10.7} - 70) + 1.5A_p \\ m &= 27 - 0.012 \times (h - 200) \\ H &= T/m \\ \rho &= 6 \times 10^{-10} \times e^{-(h-175)/H} \end{aligned} \tag{8}$$

where h is the altitude of the spacecraft in kilometers, $F_{10.7}$ and A_p [12] are time varying properties of the atmosphere, and ρ is atmospheric density in kilogram per kilometer cubed. These properties are unknown ahead of time, making them a source of process noise for the model itself.

9.1.3 Solar and Lunar Gravity

The gravitational acceleration of the Sun and Moon were implemented using a lookup table with information pulled from the NASA Horizons website [15] for accurate positional data of both celestial bodies. The positions were chosen using a linear interpolation function to match the position to the current Julian date. This process was advantageous because it didn't require propagating extra bodies and was more accurate than any real time propagation.

9.1.4 Attitude Update Step

The derivative of the rotation rate comes from combining Euler's equation with the torque $\mathbf{M}_G$ from equation 5 to create equation 9

$$\left[{}^{\mathcal{B}}\frac{d}{dt}{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}} = \mathbb{I}_{\mathbb{G}}{}^{-1}\left(\mathbf{M}_G - {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}} \times \mathbb{I}_{\mathbb{G}}{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}\right)\right]_{\mathcal{B}} \tag{9}$$

The update to the quaternion is slightly more complicated. It first requires finding the rotation rate of $\mathcal{B}$ in $\mathcal{A}$ as in equation 10 [8].

$$
\begin{aligned}
\left[{}^{\mathcal{A}}\boldsymbol{\omega}^{\mathcal{B}}\right]_{\mathcal{B}} &= \left[{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}\right]_{\mathcal{B}} - \left[{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{A}}\right]_{\mathcal{B}} \\
\left[{}^{\mathcal{A}}\boldsymbol{\omega}^{\mathcal{B}}\right]_{\mathcal{B}} &= \left[{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}\right]_{\mathcal{B}} - {}^{\mathcal{B}}C^{\mathcal{A}} \begin{bmatrix} 0 \\ 0 \\ \frac{||\boldsymbol{v}\times\boldsymbol{r}||}{||\boldsymbol{r}||^2} \end{bmatrix}_{\mathcal{A}}
\end{aligned}
\tag{10}
$$

Using equation 10, ${}^{\mathcal{B}}\dot{\mathbf{q}}^{\mathcal{A}}$ can be calculated as in equation 11.

$$
{}^{\mathcal{B}}\dot{\mathbf{q}}^{\mathcal{A}} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \left[{}^{\mathcal{A}}\boldsymbol{\omega}^{\mathcal{B}}\right]_{\mathcal{B}} \tag{11}
$$

By combining the results of equations 10 and 11 with the sum of all disturbance accelerations, the full derivative state vector could be calculated for every step. This allowed the state to be propagated properly through time.

9.1.5 Measurements

For this satellite, we used the measurements from a GPS and star tracker with sensor noise distributions modeled by the Sentinel M-Code GPS Receiver [13] and the NST Nano Star Tracker [14]. The GPS measured position and velocity with a Gaussian noise distribution equal in all directions. The noise distributions of the star tracker are two Gaussians. One is the error in the accuracy of the angle about the boresight axis, $\tilde{\theta}_{boresight}$. The other is the error angle normal to the boresight axis, $\tilde{\theta}_{normal}$. The rotation from the actual attitude to the measured attitude was a combination of two rotations. The first was a rotation about the boresight axis, the z axis for this spacecraft, by the $\tilde{\theta}_{boresight}$. The second rotation was about a random vector, $\mathbf{n}$, in the $\mathbf{b}_1 - \mathbf{b}_2$ plane by $\tilde{\theta}_{normal}$. These successive rotations are represented by the DCM in equation 12.

$$
{}^{\mathcal{B}_M}C^{\mathcal{B}} = R_{\hat{n}}(\tilde{\theta}_{normal})R_z(\tilde{\theta}_{boresight}) \tag{12}
$$

where $\mathcal{B}_M$ is the measured body frame.

9.2 Imaging Measurement Function

9.2.1 Process of Generating Data

The process of generating the image data used for creating the measurement model can be separated into several major steps outlined below.

First, a trajectory was generated with a semimajor axis of 400 km, 20 degree inclination, and an eccentricity of 0.01. Positions were sampled at a rate of 100 per orbit over 3 orbits. After converting these positions to Latitude-Longitude-Altitude, we queried the USGS EarthExplorer API for image data [16]. Specifically, we used Landsat 8 Level 1 data for a rectangular region of side length 0.2 degrees Latitude and 0.2 degrees Longitude centered around each of the trajectory points. One important assumption that was made in order to use the positions to query data was that the angle of incidence between the vector from the camera to the Earth's surface has a zero angle of incidence.

Once we established a library of raw images, we applied a sequence of transformations and distortions to each image. First, we selected a 300 by 300 pixel region in the image centered pixel location (X_p, Y_p) with the equation for X_p given by equation 13

$$X_p = S_x \frac{\theta - B_{\theta 2}}{B_{\theta 1} - B_{\theta 2}} \tag{13}$$

where S_x is the pixel width of the image, θ is the longitude coordinate of the spacecraft position, $B_{\theta 1}$ is the left longitude bound on the image, and $B_{\theta 2}$ is the right longitude bound on the image. An equivalent calculation is used to find Y_p using the latitude coordinate of the spacecraft position and the latitude image bounds.

We went with this approach as a first pass to get data, but in the future using a tool like USGS ISIS [17] would enable a higher fidelity transformation between the 3D space and the image plane. Once the viewpoints are selected, we remap them to a 200 by 200 pixel region using homogeneous coordinates. Then, we apply a Gaussian filter [18] to the image with a standard deviation that is a function of the camera focal length f [19]. This process results in a blurred image. Then, we apply a radial camera distortion model to the blurred image based on the three camera parameters, C_1, C_2, and C_3 [20]. Finally, we execute SIFT on this distorted image and record the descriptors of all the keypoints and the position, orientation, and scale for each keypoint.

This process was repeated for each image from the original image library with a set of 1000 different sets of focal length and camera parameters. The focal lengths were chosen from a uniform distribution between 2 and 6 mm, and the camera parameters were chosen sequentially to enforce $|C_3| > |C_2| > |C_1|$. Specifically, C_3 was uniformly distributed between -20 and 20, C_2 was uniformly distributed between $-0.5C_3$ and $0.5C_3$, and C_1 was uniformly distributed between $-0.5C_2$ and $0.5C_2$.

Once we had keypoints information for each distortion of the images associated with each position, we filtered the keypoints so that each position had exactly one keypoint associated with it. Without this filtering, the measurement function would need to output a variable number of keypoints for each distortion and position combination, which would add an extra layer of complexity. We chose one keypoint to start with since it is both a simple first attempt and accounts for the fact that some images only had one keypoint. For each of the positions, we chose the keypoint that occurred in the largest number different distortions for that position. To identify which keypoints were the same among a distorted group of images based on a single position, we used a hierarchical clustering algorithm on the keypoint descriptors using the Euclidean distance between descriptors as the distance function [21]. Once the keypoint data is clustered, we chose the keypoint descriptors in the largest cluster to represent that image. If multiple keypoint descriptors from the same image and distortion were in this cluster, all but one were removed. From this cluster, we add the spacecraft position, three radial camera distortion parameters, and the camera focal length associated with the keypoint descriptors in the cluster to the set of inputs $\mathcal{X}$. The 128 element keypoint descriptor and the keypoint X and Y position, orientation, and scale associated with the keypoint descriptors were added to the set of outputs $\mathcal{Y}$. Therefore, the set of inputs to our measurement model, $\mathcal{X}$, has a dimensionality of 7 while the set of outputs from our measurement model, $\mathcal{Y}$, has a dimensionality of 132.

9.2.2 Linear Imaging Measurement Model

Given the set of image states, $\mathcal{X}$, and the corresponding set of the most observed features in the imaging state, $\mathcal{Y}$, one of the simplest means of generating a predictor to translate between a specific input and its corresponding output was to generate a linear fit function. The word linear is used lightly here because although the coefficients of the relationship between input and output are linear, the states were initially augmented into polynomial functions who's coefficients were learned through a least squares fit. The process of transforming the state and finding the least squares parameters is shown in equation 14

$$\begin{aligned} \mathcal{X}^* &= f(\mathcal{X}) \\ \mathbf{\Theta} &= \mathcal{Y}\mathcal{X}^{*T}\left(\mathcal{X}^*\mathcal{X}^{*T}\right)^{-1} \end{aligned} \tag{14}$$

where f is a transformation function, $\mathbf{\Theta}$ is the matrix of unknown parameters such that $||\mathbf{\Theta}\mathcal{X}^* - \mathcal{Y}||_2$ is minimized.

The figure of merit for measuring performance of different transformation functions, f, was the means and standard deviations of the error between the predicted and true states. In the ideal case, the mean is zero and the standard deviation is minimized. A non-zero mean represents some bias in the measurements, a factor that would negatively effect the performance of the filter. A large standard deviation will result in poorer filter performance but could still be handled by the filter through the filter's measurement covariance.

The functions chosen and compared were order 1-5 polynomials of the original state. Each term was scaled by a prespecified maximum value of every element of the state so that the condition number of $\mathcal{X}^*\mathcal{X}^{*T}$ could be minimized for as high an order as possible. However, after order 5, the condition number became too large for accurate results.

9.2.3 Neural Network Measurement Model

The neural network consisted of an input layer, a fully connected layer with 4000 neurons, a full connected layer with 132 neurons, and a regression output layer. Since the input feature set only has a size of 7, there is no need to use any kind of convolution or pooling to reduce the dimensionality of the input. The final fully connected layer size is constrained by the fact that the output has a dimensionality of 132, so the only decisions that can be made on the design of this network is the size of the first fully connected layer and the possibility for adding more fully connected layers.

Since the inputs and outputs had very different scales, the inputs were shifted and scaled such that each input had a mean of 0 and a standard deviation of 1. For the outputs, the position outputs were the pixel location in the 200 by 200 pixel image, and the orientation was an angle between $-\pi$ to π radians. Therefore, the keypoint positions were scaled to be between 0 and 1 by dividing by 200 and the orientation was scaled to be between 0 and 1 by adding π and dividing by 2π. The other elements of the output were unchanged.

The order in which the training set was given to the neural network was randomized. Through experimentation with different neural net structures and randomization of the inputs, we found that our chosen structure was able to provide estimates of the test set that were fairly unbiased.

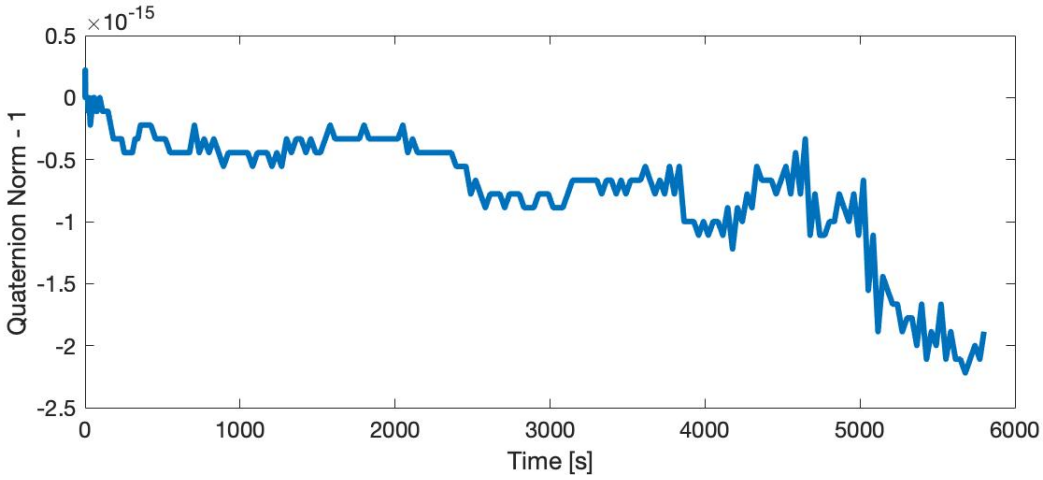

Figure 2: Normalized Quaternion difference from ideal value 1 shows precision variation 1 order of magnitude above floating point precision, verifying simulation accuracy. The relative and absolute integration tolerances are 10^{-12},

9.3 Unscented Kalman Filter (UKF)

Here, the UKF total state vector was a combination of the dynamics states and the camera parameter states with the full form $\begin{bmatrix} \boldsymbol{r}_{G/O} & {}^{\mathcal{I}}\boldsymbol{v}_{G/O} & {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}} & {}^{\mathcal{B}}\mathbf{q}^{\mathcal{A}} & c_1 & c_2 & c_3 & f \end{bmatrix}^T$.

10 Results

10.1 Dynamics Model

In order to validate the tolerance and elements of the attitude model, we check the magnitude of the quaternion in figure 2. Theoretically, the quaternion magnitude should always be 1, so the difference plotted in 2 gives the precision of the updates. Since the variation is on the order of 10^{-15}, only 2 orders of magnitude above float precision, we are confident in the step size accuracy of the results.

With no disturbances, we would expect the satellite's position to follow a perfect ellipse. In this case, the semimajor axis, eccentricity and inclination would all be constant. However, with the perturbations of this model, we would expect the orbital elements to vary with time.

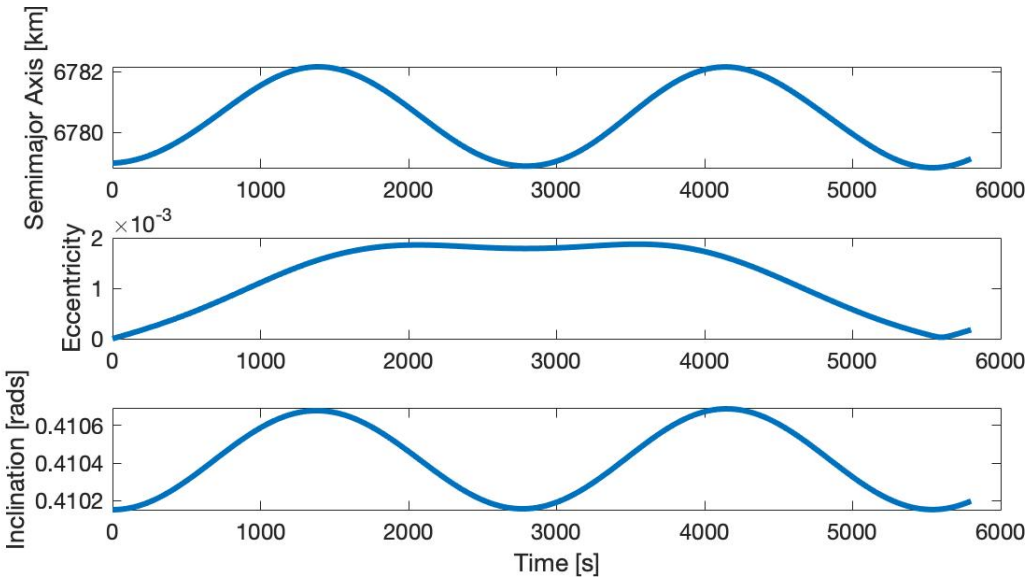

Figure 3: The variation of the orbital elements Semi-major axis, Eccentricity, and Inclination over time due to perturbations.

Figure 3 shows the semi-major axis, eccentricity, and inclination of the satellite over one full orbit. Each of these orbital elements change in time due to the perturbing forces of the moon, sun, atmospheric drag and non-spherical earth.

The satellite's energy is not conserved primarily due to the influence of drag. Figure 4 shows the energy of the satellite as a function of time through one orbit. As expected, the energy decreases throughout the orbit. The small oscillations within the total energy decrease are caused by the decrease in atmospheric density as the the semi-major axis oscillates in the orbit.

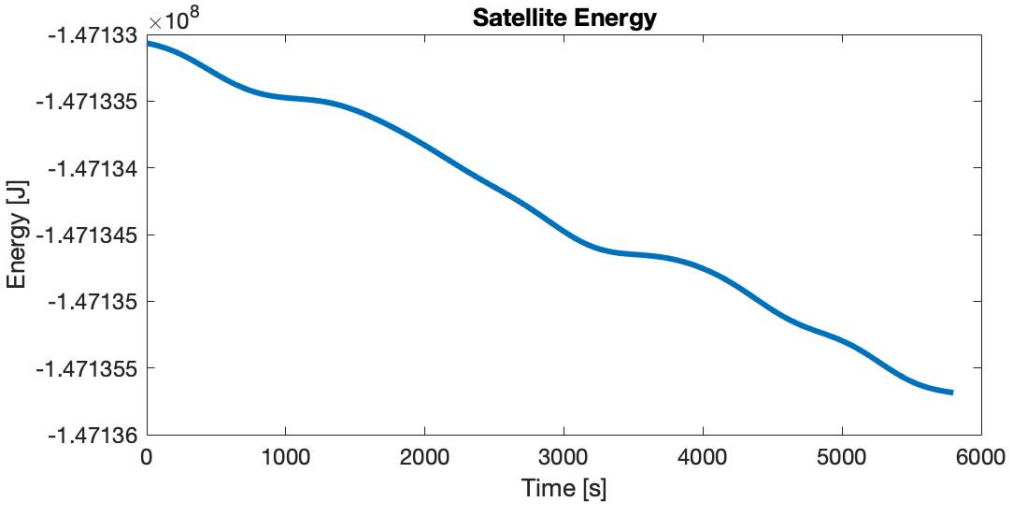

Figure 4: Satellite's total energy decreases in the presence of atmospheric drag, the moon, and non-spherical gravity.

The effects of the gravity gradient are displayed in figure 5 where the angle between the satellite boresight and the radius vector is plotted over an orbit for three cases. In all three cases, the satellite's initial state is pointing along its boresight axis antiparrallel to the radius vector. In one case, gravity gradient is turned off so this angle linearly increases to π an then back to 0 over the course of the orbit. With gravity gradient turned on, the angle is shown for both a minor axis spinner and major axis spinner. As expected, the minor axis spinner is torqued toward its radius vector strong enough that it begins to oscillate about that vector. The major axis spinner is unstable in this configuration and points the boresight away from radius vector even faster than if there was no gravity gradient.

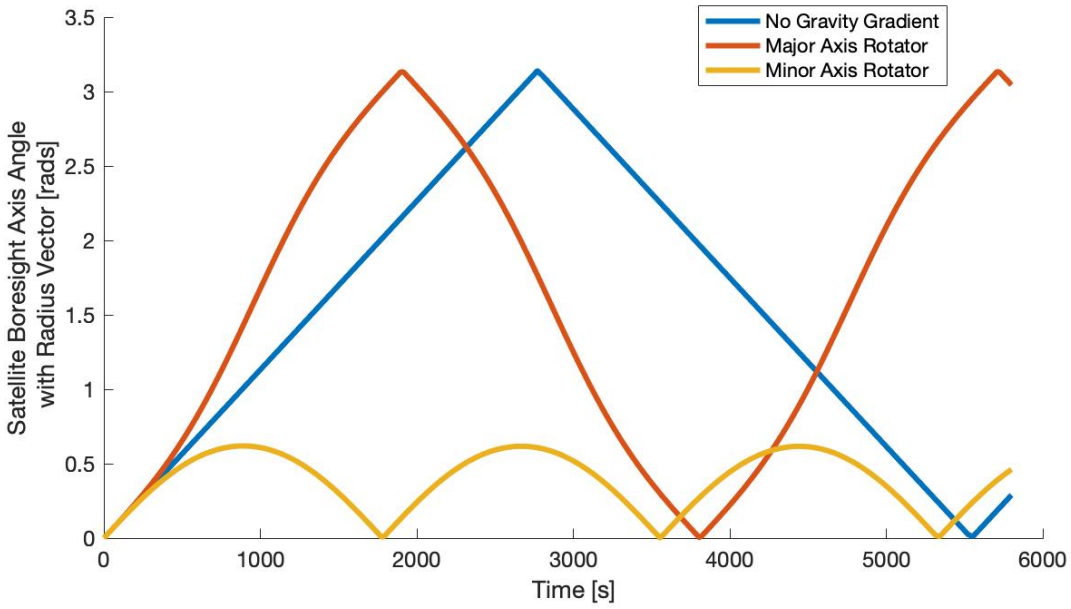

Figure 5: The angle the satellite boresight makes with its radius vector varies over the course of an orbit. The variation of angle with time is different for the minor axis rotator and major axis rotator as well as the case without a gravity gradient torque.

10.2 Comparison of Measurement Functions

The models were evaluated on two different training and test set combinations. The first set trained on images from all satellite locations and a subset of distortions and tested on all satellite location but different distortions. This test was an evaluation of the measurement model's ability to predict the keypoints of new distortions. The second combination was trained on all but one of the satellite locations and a subset of distortions with testing exclusively on the satellite location excluded during training with a mix of new and old distortions. This test evaluated the measurement model's ability to predict keypoints of new satellite locations with source images that the model has not seen before.

The first metric used to evaluate our models was the normalized inner product between the true keypoint descriptor and the predicted keypoint descriptor. A plot of this metric for both measurement models is given in figure 6 for the new distortions case. The case with a new satellite location is given in figure 7. If the keypoint descriptors are the same, then we would expect

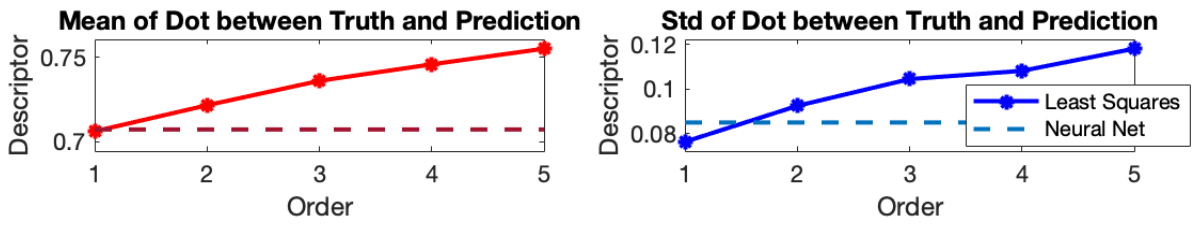

Figure 6: New test **distortion** mean and standard deviation of the dot product between the predicted keypoint descriptor and the true keypoint descriptor for both the linear model and neural network. The linear model is shown as a solid line as a function of the order of terms used and the neural net is shown as a dashed line.

the dot product between the two keypoint descriptors to be close to 1. As a result, this metric measures the similarity between two descriptors. Both the linear model and neural network had approximately the same performance in this metric. Higher orders of the linear model had slightly higher, and therefore better, mean. However, higher orders of the linear model also had larger standard deviation for this metric. The new satellite location set performed slightly better than the new distortion set with a significantly smaller standard deviation. This trend suggests that the spacecraft position heavily influences the training of the keypoint descriptor prediction. However, this method only tested on a single unknown satellite location. After running a similar procedure on more satellite locations, it became clear that the mean was heavily dependent on which satellite location was chosen to test on, suggesting that any accuracy was not repeatable.

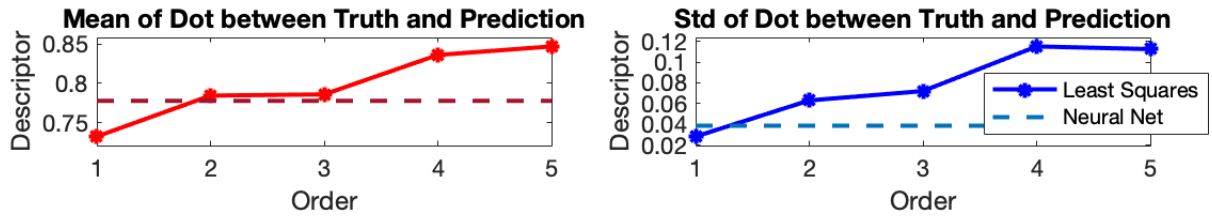

Figure 7: New test **satellite position** mean and standard deviation of the dot product between the predicted keypoint descriptor and the true keypoint descriptor for both the linear model and neural network. The linear model is shown as a solid line as a function of the order of terms used and the neural net is shown as a dashed line.

The second set of metrics used to evaluate our models was the mean and standard deviation of the error in keypoint position, scale, and orientation. Since the UKF relies on having measurement noise with zero mean error, we want the mean error for these metrics to be close to zero. A plot of these metrics for both measurement models for the new distortion case is given in figure 8 and for the new satellite position case in figure 9. Focusing first on the new distortion set. In the X and Y keypoint position, the mean position errors for the linear model are independent of the model order, and the standard deviation of the position errors decreases with model error. The neural network performance is similar to the linear model in the mean position error but is worse than the linear model in standard deviation. In both keypoint scale and orientation, the neural network achieves

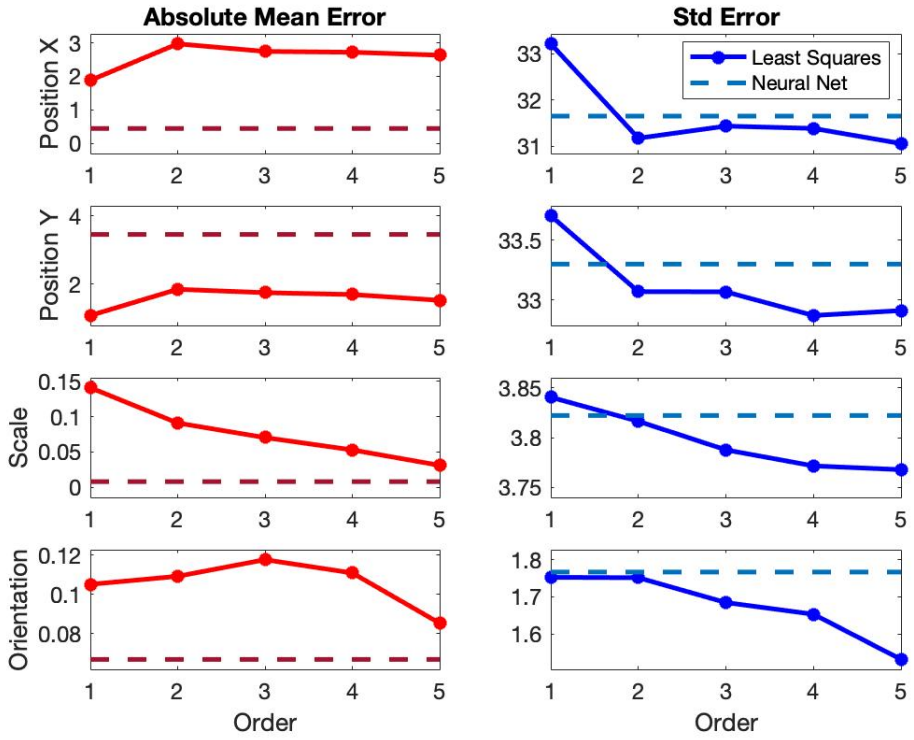

Figure 8: New test **distortion** mean and standard deviation of the error keypoint position, scale, and orientation for both the linear model and neural network. The linear model is shown as a solid line as a function of the order of terms used and the neural net is shown as a dashed line. The position error is given in pixels and the orientation error is in radians.

better performance in the mean error but worse performance in standard deviation as compared with the linear model. Additionally, increasing the order of the linear model also improves the keypoint scale and orientation mean error and error standard deviation. These trends suggest that using a higher order linear model leads to overall better accuracy, and the choice between the neural network and linear model is a trade off between better mean error and better standard deviation of error.

The new satellite location set tells a very different story. Since this image was not seen during training, the expected position was very far away from the truth. This difference caused a mean error in X of between 20 and 30 pixels for the linear model and 15 pixels for the neural network. Both approaches perform better in Y but still show significantly more error than with the new distortion set. Errors in scale and orientation are over an order of magnitude worse in mean but have smaller standard deviations than the new distortion set. This large difference in results when excluding a satellite location suggests that the measurement function can become highly biased.

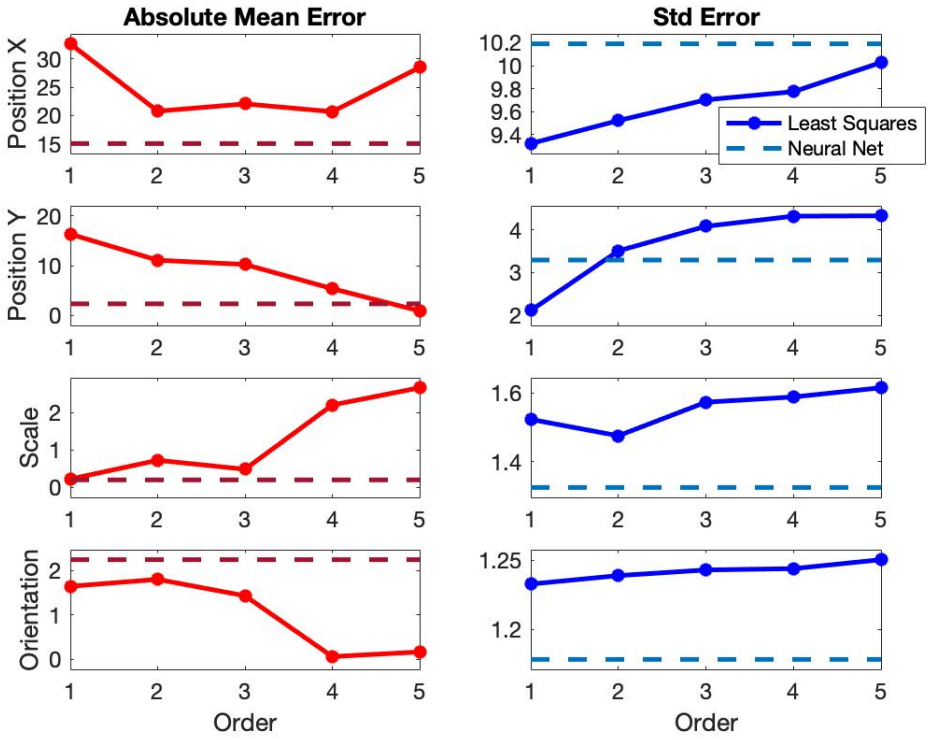

Figure 9: New test **satellite position** mean and standard deviation of the error keypoint position, scale, and orientation for both the linear model and neural network. The linear model is shown as a solid line as a function of the order of terms used and the neural net is shown as a dashed line. The position error is given in pixels and the orientation error is in radians.

11 Conclusions and Recommendations

We evaluated two methods of predicting SIFT keypoints from satellite dynamics and camera states. While we showed that these methods had close to zero mean error when they had been trained with all available original images, the measurement function had highly biased errors when the testing set included satellite positions that were not in the training set. There were also a variety of decisions made here whose influences were not explored with the final results, one of which is the decision to only select one keypoint for each satellite position. A true exploration into prediction of sets with variable number and variable contents would require the use of finite set statistics.

Overall, these results imply that the data set we are approximating is complex and difficult to capture. It likely requires a combination of a more complicated measurement function and a larger sample of data to properly represent this data mapping as a function.

12 Individual Contributions

12.1 Nathaniel Kinzly

1. Image data generation (Downloading images/distorting images/getting keypoints)

2. Neural network measurement model

12.2 Samuel Feibel

1. Dynamics Model
2. Least-Squares Measurement Model
3. Manipulating Keypoint Output from Image Generation

13 References

[1] Julier, S., &; Uhlmann, J. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3), 401-422. doi:10.1109/jproc.2003.823141

[2] Ponomareva, K., Date, P., Wang, Z. (2010). A New Unscented Kalman Filter with Higher Order Moment-Matching. *Proceedings of the MTNS*, 1609-1613.

[3] Moral, P. D. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus De L'Académie Des Sciences - Series I - Mathematics*, 325(6), 653-658. doi:10.1016/s0764-4442(97)84778-7

[4] Monica, V. M. and Nigel, K. G. J. , "Object tracking based on Kalman filter and gait feature extraction," 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2017, pp. 1-5, doi: 10.1109/ICISC.2017.8068677.

[5] Velastin, S. A., &; Xu, C. (2007). Image Feature Extraction Using a Method Derived from the Hough Transform with Extended Kalman Filtering. Advances in Image and Video Technology Lecture Notes in Computer Science, 191-204. doi:10.1007/978-3-540-77129-6_20

[6] Lowe, D. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. doi:10.1109/iccv.1999.790410

[7] Savransky, D. (2020) MAE 6720 Lecture Notes

[8] Savransky, D. (2019) MAE 4060 Lecture Note

[9] ode113. (n.d.). Retrieved December 21, 2020, from https://www.mathworks.com/help/matlab/ref/ode113.html

[10] *New Table Containing Conventional Gravity Coefficients.* www.csr.utexas.edu/publications/statod/TabD.3.new.txt.

[11] *Satellite Orbital Decay Calculations. Australian Government Bureau of Meteorology*, www.sws.bom.gov.au/Category/Educational/Space%20Weather/Space%20Weather%20Effects/SatelliteOrbitalDecayCalculations.pdf.

[12] "Space Weather Data Documentation." *CelesTrak*, www.celestrak.com/SpaceData/SpaceWx-format.php.

[13] "Sentinel M-Code GPS Receiver." *General Dynamics*, https://gdmissionsystems.com/products/communications/spaceborne-communications/spaceborne-gps-receivers/sentinel-m-code-gps-receiver.

[14] "NST 3 Nano Star Tracker." *CubeSatShop*, https://www.cubesatshop.com/product/nst-3-nano-star-tracker/.

[15] "HORIZONS Web-Interface." Edited by Alan Chamberlin, *NASA*, NASA, ssd.jpl.nasa.gov/horizons.cgi.

[16] USGS - U.S. Geological Survey. (n.d.). Earth Explorer API. Retrieved December 21, 2020, from https://earthexplorer.usgs.gov/inventory/

[17] USGS ISIS. (n.d.). Retrieved December 21, 2020, from https://isis.astrogeology.usgs.gov/Isis2/isis-bin//isis_overview.cgi

[18] Imgaussfilt. (n.d.). Retrieved December 21, 2020, from https://www.mathworks.com/help/images/ref/imgaussfilt.html

[19] Mannan, F., &; Langer, M. S. (2016). What is a Good Model for Depth from Defocus? *2016 13th Conference on Computer and Robot Vision (CRV)*. doi:10.1109/crv.2016.61

[20] Tang, Z., von Gioi, R. G., Monasse, P., Morel, J. A Precision Analysis of Camera Distortion Models. *IEEE Transactions on Image Processing, Institute of Electrical and Electronics Engineers*, 2017, 26 (6), pp.2694 - 2704. ff10.1109/TIP.2017.2686001ff. ffhal-01556898ff

[21] Clusterdata. (n.d.). Retrieved December 21, 2020, from https://www.mathworks.com/help/stats/clusterdata.html

[22] Markley, F. L., &; Crassidis, J. L. (2016). *Fundamentals of Spacecraft Attitude Determination and Control*. New York, New York: Springer.

[23] Savransky, D. (2019) "Sub-Pixel Inter-Satellite Imagery Cross-Calibration via Image Decomposition and Dynamic Filtering"