RPO Maneuver Detection from Pixel Space using Deep Learning

Emily Gerber
Ten One Aerospace

Michael Mercurio, Jason Crane, Christopher Roscoe, Jason Westphal
Ten One Aerospace

ABSTRACT

When two or more satellites are operating in close proximity, it is imperative that the states are well known and any deviation from the predicted state, as the result of a maneuver, is quickly identified and quantified to evaluate any potential danger of collision or operational impediment. In the case where Clohessy-Wiltshire-Hill (CWH) assumptions hold and a simple along track delta-V is applied to the chaser spacecraft, the analytic relation between geometric parameters of the co-planar trajectory is well characterized in academic literature. As the relative motion between satellites becomes more complex and complicating factors such as eccentricity and orbital perturbations are introduced, the need to characterize these classes of maneuvers becomes even more important. In this work, we reformulate the analytical approaches for the simplest CWH cases but from the perspective of relative pixel spatial-temporal geometry gathered by a ground-based optical sensor observing maneuvers for closely spaced objects (CSOs) in Geostationary orbit [3, 2, 8, 7, 5, 4]. Inevitably, the analytic solutions break down as the viewing geometry and the orbit parameters vary , hence, we examine if simple first-order delta-V estimates can be obtained. Once this class of analytic and first-order solutions from the pixel-spatial-temporal/ground-based-sensor framework are obtained, we compare them with an alternative data-centric machine learning-based approach to the problem. We apply deep learning techniques to train and validate regression and classification models to estimate maneuver characteristics, including delta-V, for increasingly complex orbits and RPO scenarios. The data used characteristics, including delta-V, for increasingly complex orbits and RPO scenarios. The data used to train and validate models include variations of multiple RPO scenarios such as V-bar hop and R-bar station keeping using our validated RPO simulation capability. The results of the simulations include sets of stacked images that characterize the maneuver. These data are used to train deep learning regression and classification models to estimate the delta-V and type of maneuver, respectively. Results from the deep learning models are validated against the simplified cases where analytic approaches hold, and regression accuracy of the model outside simplified cases are readily estimated since applied delta-V's in each simulation case is known. The goal of this work creates a "secondary path" for ground-sensors to rapidly characterize a maneuver using a well-trained deep neural network model based solely on the raw observations of pixel data from their sensors, without the need for first going through post-processing and classical orbit determination techniques. The value of these results will be timely capture of maneuver events to support both safe space operations and rapid task prioritization to support custody maintenance.

1. MATHEMATICAL FORMULATION

First, the analytical solution will be derived in the CWH frame and then the same distance measurements will be found from the pixels transformed based on the geometry and converted into a distance based on the

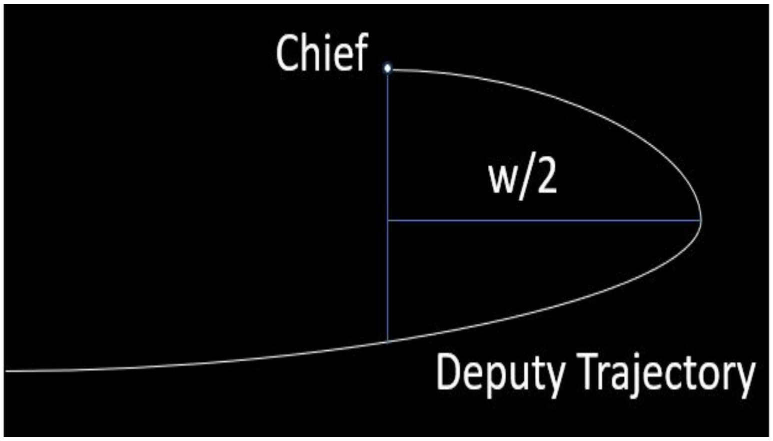

Fig. 1: Notional Trajectory Trace From Maneuver

camera properties and radial distance. Figure 1 shows the expected geometry that an observer would expect for looking perpendicularly at an along-track maneuver.

To find the magnitude of the Δv for a maneuver we start with [2] which looks at teardrop orbits. If we assume the peak of the teardrop is the same altitude as the chief (ie D = 0, Eq: (2)), then it follows the relative velocity at that peak would be identical to the delta V applied to enter the trajectory from a stationary 'string of pearls' position at the peak.

$$w = -2a_e \sin(\cos^{-1} - \frac{3X_d}{2a_e}) - 3X_d(\cos^{-1} - \frac{3X_d}{2a_e}) \tag{1}$$

$$D = X_d + \frac{a_e}{2} \tag{2}$$

If we assume $D = 0$, then we can solve for a_e and $X_d$in terms of w.

$$X_d = -\frac{a_e}{2} \tag{3}$$

$$a_e = -2X_d \tag{4}$$

Plug into equation 1 :

$$X_d = w(2.0940278493) \tag{5}$$

$$a_e = w(-4.18805569861) \tag{6}$$

From [3] we can look at the equations of motion and solve for the delta-v with the derived a_e and X_d.

$$\dot{y} = a_e n \cos(\beta) - \frac{3}{2} n X_d \tag{7}$$

Where β is the location in the ellipse, $\beta = \pi$ is the top of the ellipse. n is the mean motion of the non-maneuvering object.

Plug the a_e and X_d equations into $\dot{y}$ and solve for the delta-v.

$$\dot{y} = w(-4.18805569861)n(-1) - \frac{3}{2} n w(2.0940278493) \tag{8}$$

Simplifying and putting in terms of delta-v :

$$\Delta v = wn(1.04701392465) \tag{9}$$

With this, we can measure the distance of the width of the ”hook” created from the maneuver and estimate the applied along track delta-v. Often the width of the hook is measured as $\frac{w}{2}$ so that needs to handled appropriately.

Depending on the viewing angle, some measurements may be easier to observe than others. [7] provides a similar derivation for different maneuver/orbit types. For a V-bar hop with the height and width of the resulting shape are known, you can solve for the delta-v applied in the along track direction.

$$\Delta v = \frac{\Delta hw}{4} \tag{10}$$

To find the measured distances from pixel space, each image of the stacked images are transformed based on the current geometry between the observer and target RSOs. However, this transformation is only valid for small angles, as the angles grow larger the distance is harder to extract accurately to get an accurate maneuver magnitude estimate.

Then the pixels can be measured and converted into a real world distance :

$$X = \frac{uZ}{f} \tag{11}$$

Where u are the number of pixels, Z is the radial distance and f is the focal length. This is assuming no distortion in the camera lens. With this combination of measuring and converting the measurable distances, the maneuver can be estimated analytically, however it only applies to a small set of maneuvers and observer-target geometries. As the orbit deviates from circular, the ability to accurately measure w decreases. With limited applicability based on the observer-RSO geometry and orbit regimes, a machine learning approach was employed to show a wider applicability for different observer-RSO geometries.

2. DATA GENERATION

In order to generate adequate data for training, tuning, and testing the deep learning model, large data sets were generated. Using the CU Boulder Basilisk simulation engine [6], an external module was built to apply the delta-v and create realistic stacked images in a monte-carlo like framework.

The chief and deputy orbits were defined to be in a geosynchronous orbit at -156 degrees longitude. The sensor is modelled after an optical camera located on Maui, HI. The chief, deputy, and sensor had the same starting point for each series of data sets. Future improvements will include larger variations of the observer-target geometry and chief/deputy starting orbits. An along-track or radial delta-v was then applied from a uniform distribution. The satellite positions are propagated using a high fidelity propagator. At a specified cadence, an image of the chief and deputy was simulated using the geometry and given lighting conditions. The observer camera is oriented such that the bore sight is aligned with the chief and the deputy moves relative to the chief. The image is simulated using OpenCV (Open Source Computer Vision Library), an open-source computer vision and machine learning software library designed for real-time image and video processing. Each of the RSOs are modelled as a point-spread function (PSF) centered on the pixel location of the object. Background noise, shot noise, and hot pixels were included in the image. Then for each of the images that were generated, they were stacked by adding all of the images together, minus the chief, due to high pixel counts over many stacked images. The final image is saved as a PNG file and a text file with the applied delta-v is saved.

General Mission Analysis Tool (GMAT) is an open-source space mission analysis and design software developed at NASA's Goddard Space Flight Center to provide a flexible platform for mission planning, optimization, and trajectory analysis [1]. GMAT was used to model the orbits for the sensor, chief, and deputy shown in Figure 2.

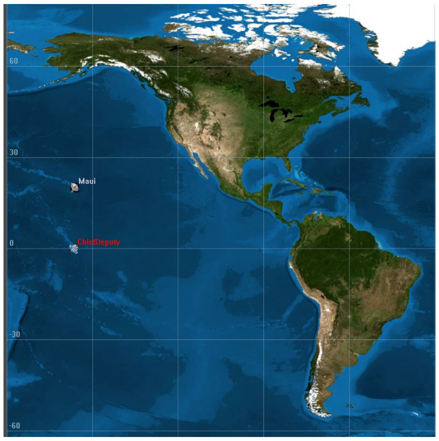

Fig. 2: RPO Scenario Modeling

Figures 3, 4, 5 show different maneuver magnitudes and the unique traces that are created between the relative motion between the chief and deputy observer by the sensor for an along track maneuver .

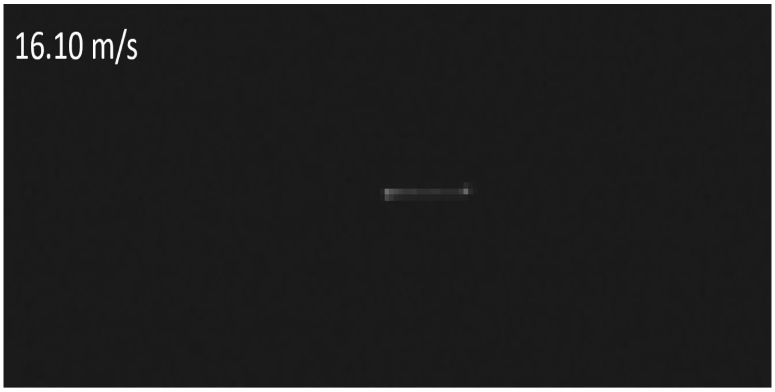

Fig. 3: Along-Track Maneuver (Small)

Due to the viewing angle of the sensor being at a higher latitude, the hook is not widely distinguishable in the stack trace of the images. This shows again how the analytical solution breaks down based on the observer-chief geometry and brings the need for a machine learning model that can be trained to still determine the correct maneuver. The direction of the burn is even evident in the direction of the hook. As the maneuvers grow larger, they will eventually go out of the field of view of the sensor and data will be lost, that can cause some issues in the machine learning model, which is addressed in the Results and Analysis section.

Figures 6, 7, 8 show different maneuver magnitudes and the unique traces that are created between the relative motion between the chief and deputy observer by the sensor for radial maneuver.

Similarly, the distinct measurable shapes are hard to distinguish in pixel space. For the smaller maneuvers,

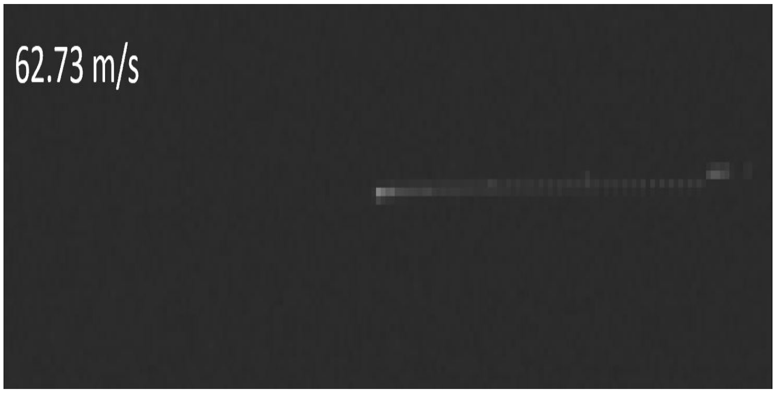

Fig. 4: Along-Track Maneuver (Medium)

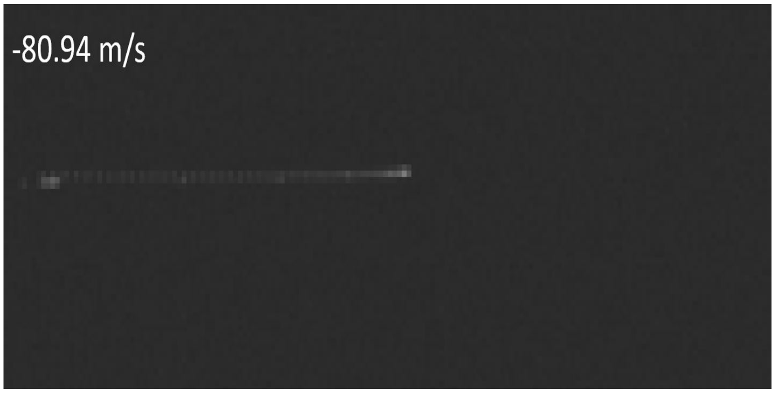

Fig. 5: Along-Track Maneuver (Large)

it is even hard to determine that a maneuver has occurred between the closely spaced chief and deputy. However, each maneuver still creates a distinct shape that is trainable for a machine learning model.

3. MACHINE LEARNING

The inputs into the deep learning model is the singular png image of all of the stacked images. The image goes through a pre-processing where the image is converted to greyscale and then converted into a binary image. Additionally, no metadata is included in the model training at this time, but will be in future work. For example, for training for a wider variety of sensor orbits, the metadata can include a radial or inertial distance to the chief or mean motion between the two orbits between the sensor and chief. The dataset is divided into a training and testing datasets.

The deep learning model starts with a 2-dimensional convolution on the input data, which is commonly used for feature extraction for images. We use RELU activation, considered standard practice for neural networks due to their simplicity, empirical performance, and ease of training compared to tanh/sigmoid units. The next layer is a 2D Max Pooling layer performs 2-dimensional max pooling to downsample the spatial dimensions of input data, retaining the maximum value within each pooling region. The data is then flattened from a 2D array to a single column vector and does dropout, which randomly sets inputs to zero to prevent over-fitting. The next layers are repeated twice that start with a Dense, Batch Normalization, Activation, and Dropout layers. The Dense layer implements a fully connected neural network layer where each neuron is connected

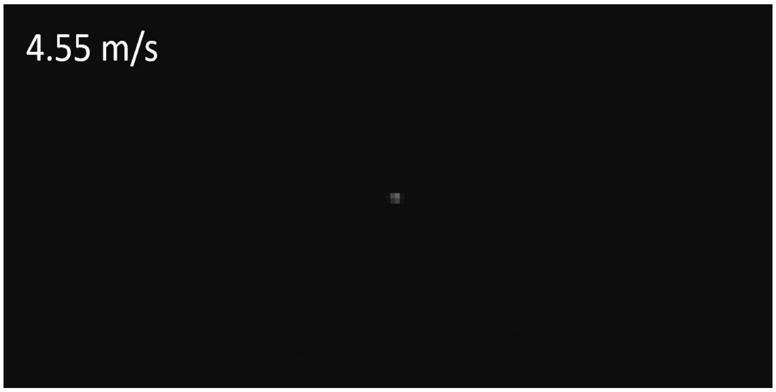

Fig. 6: Radial Maneuver (Small)

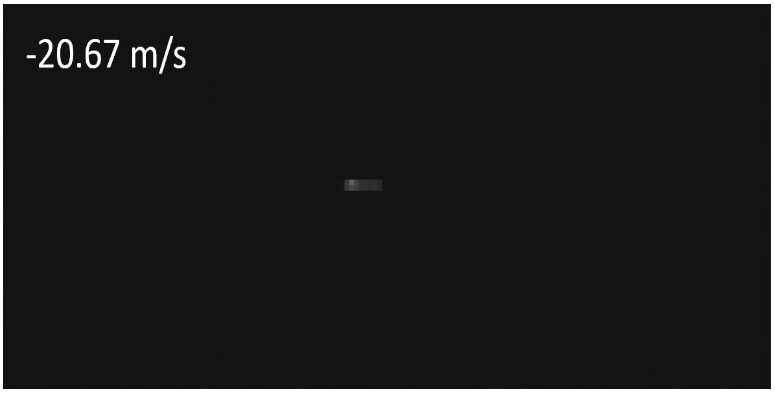

Fig. 7: Radial Maneuver (Medium)

to every neuron in the previous layer, allowing for complex learned transformations of input data. The Batch Normalization layer normalizes the activations of a neural network's previous layer across a mini-batch of data, aiding in faster training and improved generalization. Next is another Activation layer, but instead of a RelU activation function, TanH applies the hyperbolic tangent activation, mapping input values to a range between -1 and 1, often used for introducing non-linearity in neural networks. Finally, followed by another Dropout layer to avoid over-fitting. The last step is another dense layer and the output is a single estimate of the delta-v of the maneuver conducted by the chief.

This deep learning model architecture was chosen through some trial and error and experimenting with what combination of layers seemed to produce the best and consistent results. The hyperparameters within the model were also chosen through starting with approximate initial guesses and manually searching through a range of the hyperparameter space until settling on reasonable values within the model.

4. RESULTS AND ANALYSIS

To generate the data sets, for each maneuver and observer location between 500-1,000 images and metadata were generated. Each data set consists of 72 stacked images of the chief and deputy at a 10 min cadence for 12 hours. The applied delta-v varied from [-100,100] m/s in a uniform distribution. The camera had 1600 x 1200 pixels and a focal length of 30 mm and pixel pitch of 0.012 mm.

After the data was generated it went through the pre-processing described in the previous section and then

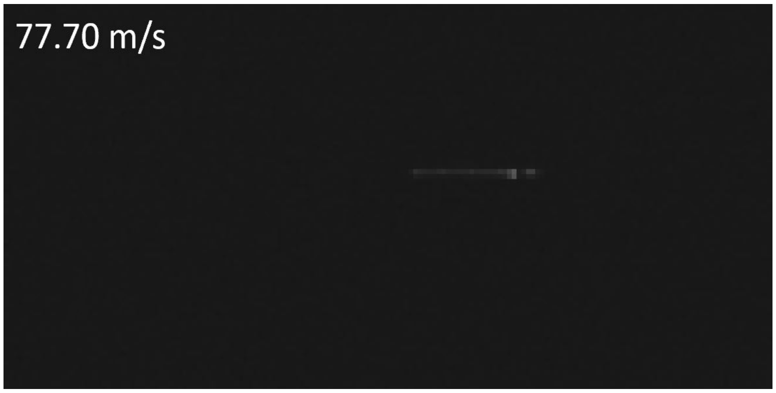

Fig. 8: Radial Maneuver (Large)

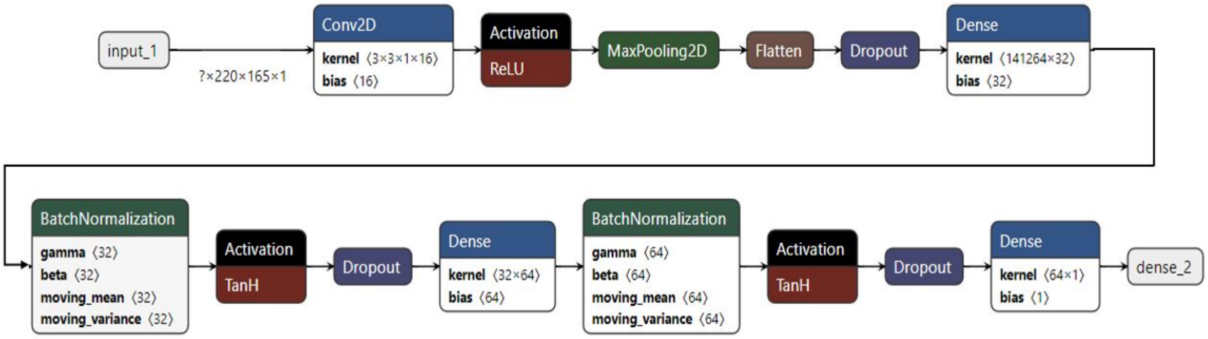

Fig. 9: Deep Learning Model

was divided into training and testing sets. Then the deep learning model was trained on the training data set. Some time and effort went into coming up with reasonable hyperparameters and model structure for each use case. Once the model was trained, the model was evaluated using the testing data set.

The first data set used for analysis was the along-track burn. The mean of burn magnitude error was 14.43% and the median burn magnitude error was 4.37%. The percent error is much larger for the smaller magnitude burns and the magnitude of the burn error is larger for the higher burns. The results can be seen in Figures 10 and 11.

At around 80 m/s, there is a change in the slope of the error. This is due to the loss of information for the deputy going out of the field of view for the sensor. This leads to the estimated maneuver being smaller than the actual maneuver and creates the negative sloped line in the error. A way to potentially resolve this in the future is to include a time element. The model is currently agnostic to time, but if a time element was included in a metadata and given to the model, the model could potentially use that to distinguish the deputy that leaves the field of view quicker than another and help determine the maneuver magnitude. For the along-track case, the error is generally less than 20 m/s, as shown in Figure 12. This result can be improved by adding more data samples for training.

The second data set used for analysis was a radial burn. The mean of burn magnitude error was 15.51% and the median burn magnitude error was 12.08%. The percent error is much larger for the smaller magnitude burns and the magnitude of the burn error is larger for the higher burns. The results can be seen in Figures

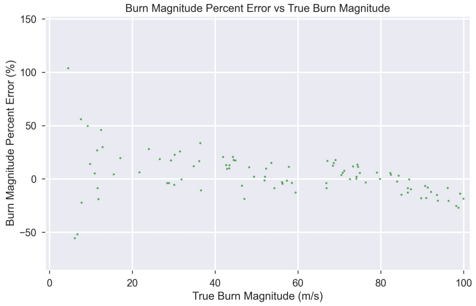

Fig. 10: Along-Track - Percent Error

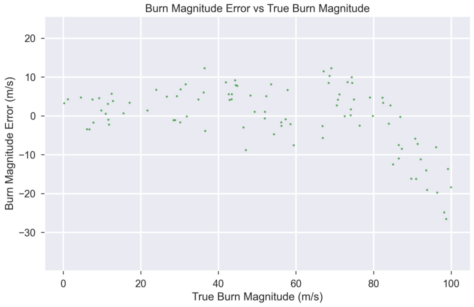

Fig. 11: Along-Track - Burn Magnitude Error

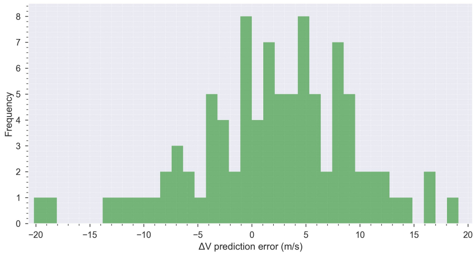

Fig. 12: Along-Track- Histogram

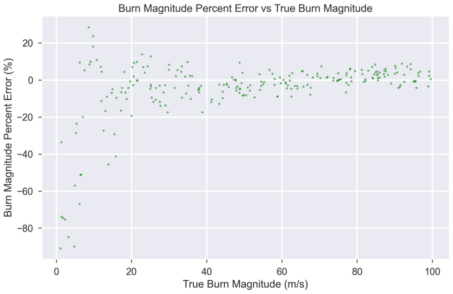

Fig. 13: Radial Burn - Percent Error

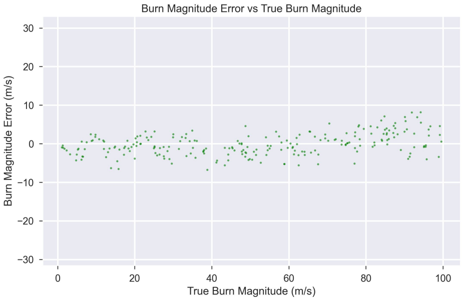

Fig. 14: Radial - Burn Magnitude Error

13 and 14.

For the radial case, the error is generally less than 10 m/s, as shown in Figure 15.

5. CONCLUSION AND FUTURE WORK

In conclusion, a proof of concept data generation and deep learning model were created and demonstrated their applicability in estimating the magnitude of a maneuver for different types of maneuver from a ground based observer. Additionally, a brief analytical solution was found but can degrade significantly based on the observer-chief geometry and orbit parameters. The overall results performed well with the limited data and test cases.

This solution provides a secondary path from the formal orbit determination for maneuver estimation to be used to effectively and efficiently estimate the magnitude of a manuever of two closely space-objects and help maintain safety of flight for both objects.

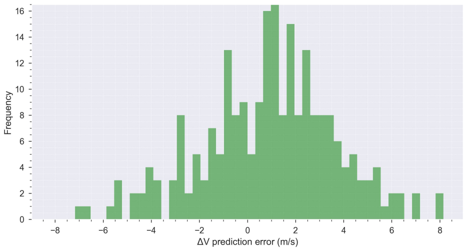

Fig. 15: Radial - Histogram

Table 1: Mean/Median Burn Magnitude Error

Use Case	Mean Burn Magnitude Percent Error	Median Burn Magnitude Error
Along-Track	14.43%	4.37 %
Radial	15.51%	12.08%

In future work, there are lots of ways to expand and improve on the work already completed. Starting with training a single model for all different types of observer-chief orbit types and viewing angles. This will also necessitate the inclusion of mixed data in the form of metadata that will include information about the relative geometry to help the model train on the different orbit types. Additionally, there are realism improvements that can be made to the image generation process, such as changing the diffraction model and including exclusion limits with the sun. All of these improvements will be able to take the current proof of concept model and enhance it to the next level of maneuver estimation using space-based images.

6. ACKNOWLEDGEMENTS

I would like to thank my co-authors for all of their hard work and support in writing this paper. I would also like to thank Kyle Nave (Ten One Aerospace) for always being able to answer questions and going above and beyond.

7. REFERENCES

[1] General mission analysis tool (gmat). `https://gmat.gsfc.nasa.gov`, 2023.

[2] T. A. Lovell and D. L. Brown. Impulsive-hover satellite trajectory design for rendezvous and proximity operation missions. In *AAS Rocky Mountain Guidance, Navigation, and Control Conference*, 2007.

[3] T. A. Lovell and M. V. Tollefson. Calculation of impulsive hovering trajectories via relative orbit elements. *Adv. Astronaut. Sci.*, 123:2533–2548, 2006.

[4] E. Prince and R. G. Cobb. Optimal guidance for relative teardrops with lighting and collision constraints. In *2018 AIAA Guidance, Navigation, and Control Conference*, 2018.

[5] Y. Rao, J. Yin, and C. Han. Hovering formation design and control based on relative orbit elements. *J. Guid. Control Dyn.*, 39(2):360–371, Feb. 2016.

[6] Hanspeter Schaub. Basilisk - multi-body simulation toolkit. `https://hanspeterschaub.info/basilisk/`, 2023.

[7] D. Woffinden. Angles-only navigation for autonomous orbital rendezvous, Dec. 2008.

[8] D. C. Woffinden. On-orbit satellite inspection: Navigation and δv analysis. 2004. Accessed: Feb. 23, 2023. [Online]. Available: `https://dspace.mit.edu/handle/1721.1/28862`.