1st IAA Conference on Space Situational Awareness (ICSSA)

Orlando, FL, USA

IAA-ICSSA-17-0X-XX
REAL-TIME HARDWARE-IN-THE-LOOP HAND-OFF FROM A FINDER SCOPE TO A LARGER TELESCOPE

Daniel Aguilar Marsillach[2], **Shahzad Virani**[3], **Dr. Marcus J. Holzinger**[3]

[1] *Graduate Research Assistant, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, 470-4450762, dam6@gatech.edu*

[2] *Graduate Research Assistant, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, 678-5446054, virani.shez@gatech.edu*

[3] *Assistant Professor, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, 404-3853342, holzinger@ae.gatech.edu*

Keywords: *Finder Scope, Electro-Optical Sensors, Tracking, Controls, Raven-class Telescope*

Electro-optical sensors play an increasingly important role in the SSA domain for tracking satellites and debris objects. Such sensors provide data that complement other methods, like radar based sensing, by providing a higher angular resolution, and thus improving the estimation of an object's orbit, attitude and physical properties. The acquisition of such data is invaluable for obtaining more accurate collision risk assessments and formulating improved debris mitigation efforts. The Georgia Tech - Space Object Research Telescope aims to improve detection and tracking for agile Raven-class telescopes with narrow fields of view and high angular resolutions. A secondary imaging system was used to correct the Georgia Tech - Space Object Research Telescope's pointing errors for tracking objects at high angular rates using a closed-loop controller. This paper will focus on the development and results of a real-time hardware-in-the-loop hand-off from a finder scope to a larger telescope.

1. Introduction

The purpose of Space Domain Awareness (SDA) is to provide decision-makers quantifiable and timely evidence of behavior(s) attributable to specific space domain threats and hazards.A foundational input to SDA activities are direct optical, radar, and other observations of on-orbit objects. Presently, there are an excess of 20,000 resident space objects (RSOs) larger than 10 cm in Earth orbit. Of these, around 1,100 are active spacecraft.

As the number of space objects increases with time, so too does the risk of possible collisions between these objects [1]. Past fragmentations of larger space objects have resulted in the formation of orbital debris, even when measures were taken to mitigate the possibility of such outcomes [2]. Hence, to reduce the risk of future collisions, precise monitoring and tracking of space objects and debris is required. Typically, radar or optical-based sensors are used to perform this task. Electro-optical sensors can provide information about an object's physical properties, attitude, and orbit. Hence, tracking space objects with ground-based telescopes is essential to futher understand the orbital population. This paper focuses on the development of an autonomous tracking capability for the Georgia-Tech Space Object Research Telescope (GT-SORT), which will improve GT-SORT's ability to acquire useful space object data.

Telescopes with wide fields of view, between 1°-2°, are able to track space objects with open-loop tracking using Two-Line Elements (TLEs) and commercial tracking software [3],[4]. The GT-SORT is capable of tracking Low-Earth-Orbit (LEO) and Geostationary Earth Orbit (GEO) objects. However, due to its narrow field of view (FoV), TLE errors are often larger than the GT-SORT's FoV, resulting in missed detections if used open-loop. To obtain additional pointing information, a finder scope is used. The main objective was to develop an in-house closed loop tracking capability for low-earth-orbit (LEO) objects using such a finder scope.

This paper will focus on the theory required to develop the controller, the approach and methodology used to carry the project to fruition, and a portrayal of the results obtained with the developed architecture. Three main tasks were necessary to complete this endeavor. The first, interfacing the finder scope camera and mount hardware with the observatory computer. The second, developing control algorithms to control the telescope's mount. The third, implementing object detection techniques with estimation techniques to use as state feedback for the controller.

Table 1: GT-SORT Performance

Focal Ratio	FoV (arcmin.)	iFoV (arcsec.)	Limiting Magnitude
f/6	14.23 x 11.40	0.31	12.9

Table 2: Rokinon 135 mm Lens Performance

Focal Ratio	FoV (arcmin.)	iFoV (arcsec.)	Limiting Magnitude
f/2	287.58 x 181.38	8.127	12.06

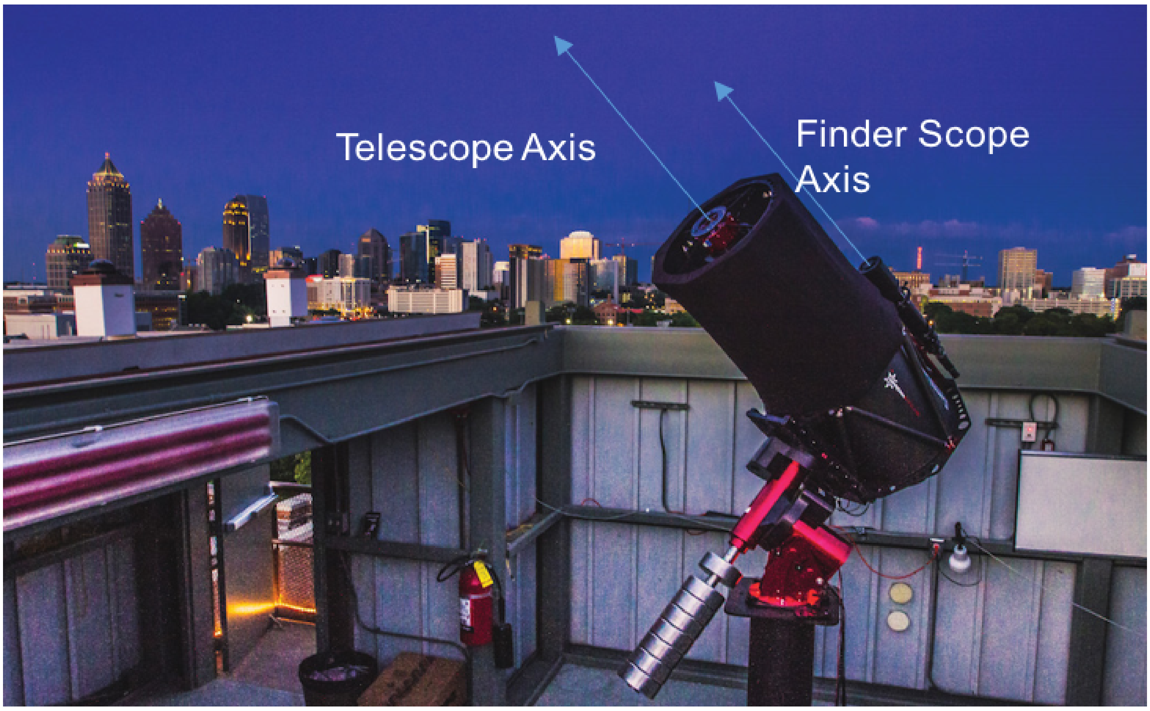

Figure 1: **GT-SORT Ready to Track Space Objects from Midtown Atlanta, USA.**

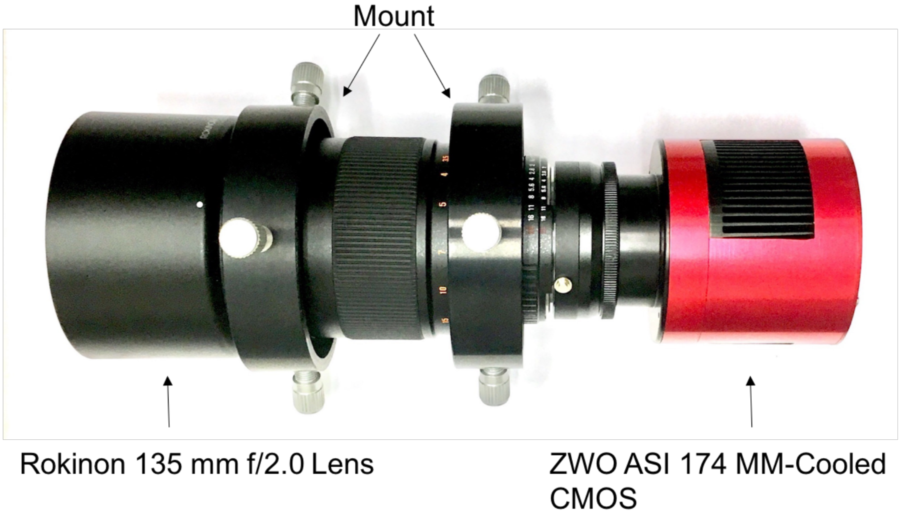

Figure 2: **Rokinon Finder Scope**

2. Theory

2.1. Dynamics and Control

This section covers the theory used to develop the controller and estimate the telescope's pointing errors with respect to the space object being tracked. The theory is tailored for a velocity (rate) controller in ra-dec state space. TSX, the software used to control Software Bisque's Paramount ME II, used with GT-SORT, offers a capability to set tracking rates. This is exploited in the development of a controller that can navigate the telescope to ra-dec positions on a particular side of the local meridian, since the Paramount ME II is a German-Equatorial mount.

The telescope's system dynamics, denoted by $\boldsymbol{x}(t)$, are modeled with the following state in the inertial ECI angular system (2 states because a rate controller is used):

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\delta}(t) \end{bmatrix} \tag{1}$$

The dynamics of the object's true trajectory are given by:

$$\dot{\boldsymbol{x}}_{\boldsymbol{O}}(t) = \begin{bmatrix} \dot{\alpha}_O(t) \\ \dot{\delta}_O(t) \end{bmatrix} \tag{2}$$

For TLE reference trajectory tracking, the telescope's dynamics are broken into the TLE nominal trajectory and the control input as:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{\alpha}_{TLE}(t) \\ \dot{\delta}_{TLE}(t) \end{bmatrix} + \begin{bmatrix} u_\alpha(t) \\ u_\delta(t) \end{bmatrix} \tag{3}$$

The TLE trajectories are the nominal ones. By controlling around these, the position errors, ϵ can be found, without any loss of generality:

$$\boldsymbol{\epsilon}(t) = \boldsymbol{x}_{\boldsymbol{O}}(t) - \boldsymbol{x}(t) \tag{4}$$

By breaking up the object's true rates into the TLE rates and some perturbation, η, the following is obtained:

$$\boldsymbol{\epsilon}(t) = \boldsymbol{x}_{\boldsymbol{TLE}}(t) - \boldsymbol{x}(t) + \boldsymbol{\eta}(t) \tag{5}$$

Taking the derivative of the above with respect to time, results in:

$$\dot{\boldsymbol{\epsilon}}(t) = \dot{\boldsymbol{x}}_{TLE}(t) - \dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{\eta}}(t) \tag{6}$$

When the telescope is tracking the TLE trajectory, the above expression than simplifies to:

$$\dot{\boldsymbol{\epsilon}}(t) = \begin{bmatrix} u_\alpha(t) \\ u_\delta(t) \end{bmatrix} + \boldsymbol{\eta}(t) \tag{7}$$

Note, the telescope's ra-dec is denoted by $\boldsymbol{x}$, the target's (space object) true ra-dec is $\boldsymbol{x}_{\boldsymbol{O}}$ and the predicted ra-dec of the target from TLEs is denoted by $\boldsymbol{x}_{\boldsymbol{TLE}}$. The majority of the error between the telescope and the object's true position are dependent on the control terms. The η term contains the inherent TLE errors. Figure 3 shows how these trajectories are seen through the finder scope's FoV, along with the true state, estimated state, and perturbation η. If the TLE's nominal trajectory and rates are tracked, then any TLE errors cause the satellite to slowly drift with respect to the optical sensor on the ground. This phenomena is evident while tracking LEO objects and less so with MEO or GEO cases.

Given the time an object is being tracked, it is assumed that ra-dec rates are well understood. Additionally, in the time interval considered, any perturbations on the object while tracking are small with respect to its period; hence $\eta(t) \approx 0$. If an object is tracked open-loop, the following error dynamics are expected:

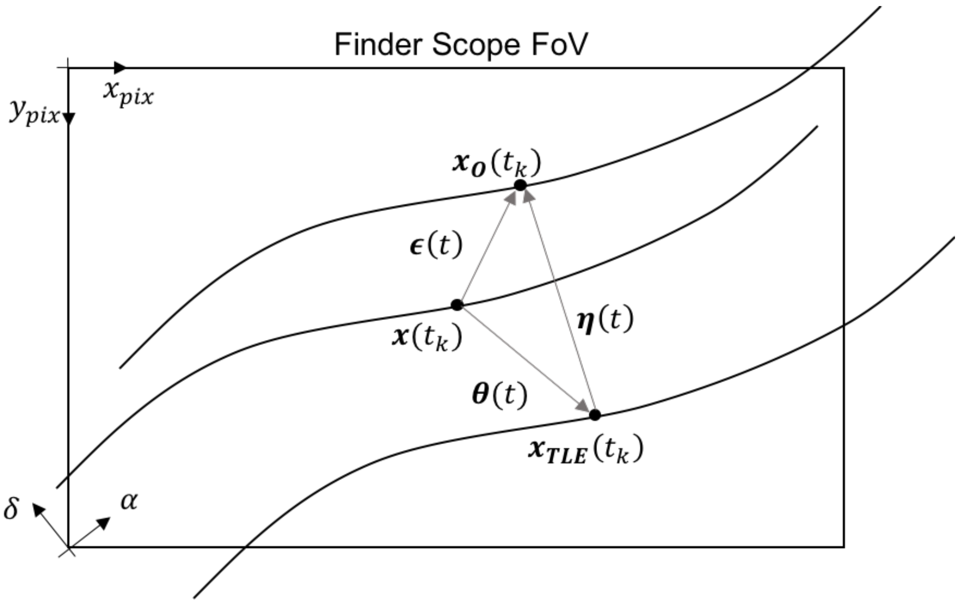

Figure 3: Trajectories as seen by the Finder Scope

$$\dot{\boldsymbol{\epsilon}}(t) \approx 0 \tag{8}$$

If there is any drift between the object and the TLE, the object detection and state estimator should capture these perturbations and feed the estimates to the controller, removing the drift. Hence, making the assumption regarding the error dynamics is valid as long as the state estimate is fed back frequently.

By changing the telescope's rate away from the nominal TLE trajectory results in the following error dynamics:

$$\dot{\boldsymbol{\epsilon}}(t) = \boldsymbol{u} \tag{9}$$

Using Equation 9, a suitable control law can be designed to correct for any telescope pointing errors. An LQ control law is implemented by solving the following optimal control problem:

$$\begin{aligned} \min_{u} \quad & \int_{t_0}^{t_0+\Delta t} \left(\boldsymbol{\epsilon}(\tau)^T \boldsymbol{Q} \boldsymbol{\epsilon}(\tau) + \boldsymbol{u}(\tau)^T \boldsymbol{R} \boldsymbol{u}(\tau)\right) d\tau + \boldsymbol{\epsilon}(t_0 + \Delta t)^T \boldsymbol{S} \boldsymbol{\epsilon}(t_0 + \Delta t) \\ \text{s.t.} \quad & \dot{\boldsymbol{\epsilon}}(t) = \boldsymbol{u} \end{aligned} \tag{10}$$

$$u \in [u_{min}, u_{max}] \tag{11}$$

Where the $\boldsymbol{Q}, \boldsymbol{R}$, $\boldsymbol{S}$ are the state, control and terminal cost matrices, respectively. Note that $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive-definite and $\boldsymbol{S}$ is positive semi-definite. Equation 12 then gives the optimal control solution [5], [6].

$$\boldsymbol{u}^*(t) = -\boldsymbol{R}^{-1} \boldsymbol{B}^T \boldsymbol{P}(t) \boldsymbol{\epsilon}(t) \tag{12}$$

The solution for $\boldsymbol{P}$ can be found via:

$$\begin{bmatrix} \dot{\boldsymbol{\Phi}} \\ \dot{\boldsymbol{Y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \\ -\boldsymbol{Q} & -\boldsymbol{A}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{Y} \end{bmatrix} \tag{13}$$

Time-invariant $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}$, and $\boldsymbol{Q}$ matrices then result in:

$$\boldsymbol{A} = \boldsymbol{0}_{2x2} \tag{14}$$

$$\boldsymbol{B} = \boldsymbol{I}_{2x2} \tag{15}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \\ -\boldsymbol{Q} & -\boldsymbol{A}^T \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{Y} \end{bmatrix} = e^{\boldsymbol{M}(t-\Delta t)} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{S} \end{bmatrix} \tag{17}$$

$$\boldsymbol{P} = \boldsymbol{Y}\boldsymbol{\Phi}^{-1} \tag{18}$$

$$\boldsymbol{P}(t_0 + \Delta t) = \boldsymbol{S} \tag{19}$$

A similar approach to the above is used for the LQ infinite-time horizon controller. This requires $(\boldsymbol{A}, \boldsymbol{B})$ to be completely controllable and $(\boldsymbol{A}, \boldsymbol{Q}^{1/2})$ to be completely observable.

$$\dot{\boldsymbol{P}} = \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} = \boldsymbol{0} \tag{20}$$

Solving the above Riccatti equation, which has a known optimal control law [5] is given by:

$$\boldsymbol{u}^* = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\epsilon} \tag{21}$$

However, effective use of the optimal controller requires accurate error estimates.

2.2. Image Processing

Image processing techniques are used to measure the telescope's attitude and pointing errors. Canny edge detection [7] failed to detect objects reliably due to the noise characteristics of the image. Hence, a matched filtering based approach (template matching) [7] was also tested with the original finder scope. Due to large processing times and the availability of a new finder scope with a larger aperture, this approach was replaced with an signal-to-noise ratio (SNR) based thresholding. Furthermore, a Gaussian blur filter was used to get accurate centroid information, whilst also reducing noise.

For SNR-based thresholding, the background noise needs to be well characterized. Additionally, in a highly light polluted area, like mid-town Atlanta where GT-SORT is located, background estimation and subtraction becomes significant due to the noise variability throughout the image. Once the effective background is subtracted, a pixel-wise SNR map can be estimated as:

$$SNR = \frac{\mathbb{E}[s_i]}{\sqrt{Var(s_i)}} = \frac{\mu_{sig}}{\sigma_{sig}} \tag{22}$$

With an SNR map, a particular SNR threshold, SNR_t, can be chosen to form the following binary image:

$$I_{x,y} = \begin{cases} 0, & I_{x,y} < SNR_t \\ 1, & I_{x,y} \geq SNR_t \end{cases}$$

This binary image is used to extract blob and region properties. The object is detected using this method and its centroid serves as raw measurements for the estimator.

2.3. Estimation

A Kalman filter is used to estimate the object's state in the finder scope's FoV [8]. Due to the integration time of the sensor, measurements are taken in discrete-time and fed to the controller, which is modelled in continuous time. Although the original telescope states are in terms of angular position, the Kalman filter estimates both the angular position and velocity. The filter incorporates the ra-dec rates as follows:

$$\boldsymbol{x}_{O,k} = \begin{bmatrix} \alpha \\ \delta \\ \dot{\alpha} \\ \dot{\delta} \end{bmatrix} \tag{23}$$

The model for the Kalman filter is based on the following:

$$\begin{aligned} \boldsymbol{x}_{O,k+1} &= \boldsymbol{\Phi x}_{O,k} + \boldsymbol{\omega}_k && (24) \\ \boldsymbol{y}_k &= \boldsymbol{H}_k \boldsymbol{x}_{O,k} + \boldsymbol{\nu}_k && (25) \\ \boldsymbol{\omega}_k &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) && (26) \\ \boldsymbol{\nu}_k &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{D}) && (27) \\ \boldsymbol{\Phi}_k &= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} && (28) \\ \boldsymbol{H}_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} && (29) \end{aligned}$$

Note, the state estimates of the object are fed into the controller at discrete intervals.

3. Methodology

First, to remove timing errors during the data acquisition phase and post processing phase of the project, the computer's clocks are synced with available atomic clocks through a Network Time Protocol (NTP) server. Then, the finder scope's boresight is manually aligned with the main telescope's optical axis.

Moreover, the telescope is controlled such that the object, circled in red in Figure 4, is brought inside the telescope's FoV, shown by the rectangle with cross-hairs. To achieve this, images are taken as the telescope tracks a space object based on the propagated state from the TLEs. As the telescope slews, stars streak in the image as a result of non-zero camera integration time. The stars in the image are registered with the inertial frame using Astrometry.net software[1] and is used to obtain estimates of the telescope's pointing and also of the object's state.

TLE errors, combined with GT-SORT pointing errors, often cause objects to appear outside of the main telescope's FoV and is illustrated by Figure 4. These cases mainly occur when tracking LEO space objects, which drift away from their TLE trajectories with time, if not updated regularly.

[1] http://www.astrometry.net

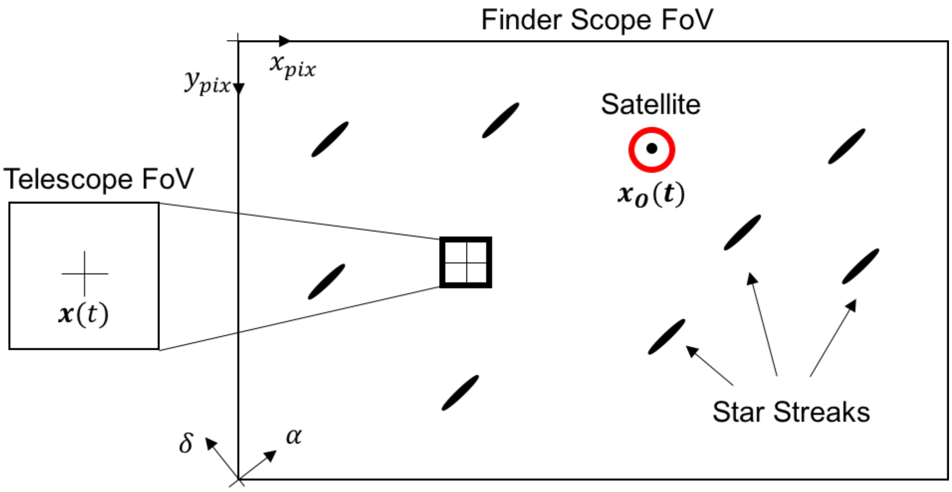

Figure 4: Graphical Representation of Problem

To correct for the pointing discrepancy in real-time, parallelized code implemented in Python is used. The sequence of operations can be seen in Figure 5. First, the TLE trajectories are computed based on the ground station's location. Then, three processes are initialized in parallel. Each stream depends on outputs of the others, but if computed in parallel (after initialization), the lag in the system is minimized and is equal to the camera's integration time. The first stream controls the finder scope camera and takes pictures for the duration of the session. The second, the image processor, is fed images from the image capture stream and processes them to compute the ra-dec position error between the telescope and the object. After the first image is taken, the image capturing and processing is done in parallel. The third proess runs the telescope mount controller, which always runs independently in the background. However, when state estimates are available from the image processor, they are incorporated into the LQ infinite time-horizon policy.

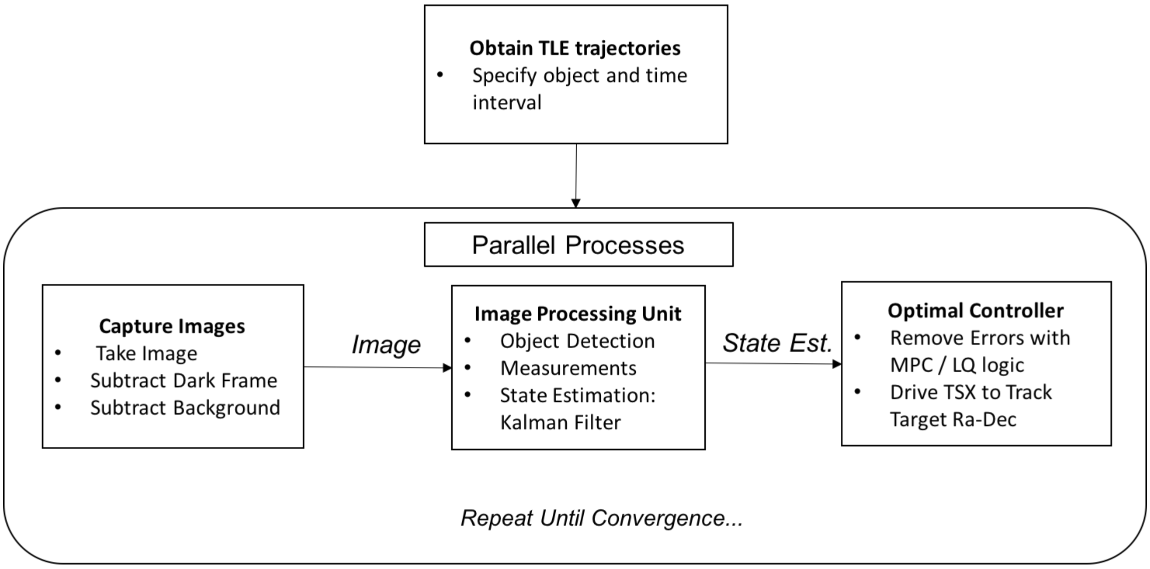

Figure 5: General Structure of Algorithm

3.1. Dynamics and Control

Once the TLE trajectories are computed a-priori, they are corrected to stop before the local meridian or start at the local meridian (depending on the trajectory of the object). This is due to the physical constraint of the German-Equatorial mount which prevents it from crossing from one side of the meridian to the other instantaneously.

The general control strategy is shown in Algorithm 1.

Algorithm 1: General Control Strategy

```
Initialization: compute required trajectories;
while Object is in groundstation's FoV do
    Obtain mount's ra-dec position;
    Compute target ra-dec position ;
    Compute error between mount ra-dec position and
      reference trajectory;
    Compute control input;
    Slew mount via TSX using input rates;
    if Error estimate received from image processor
      then
        Update reference trajectory with discrete ra-dec
          error;
    else
        Continue tracking reference trajectory;
    end
end
```

The mount's current ra-dec position is readily available from TSX and the target position is the TLE trajectory offset by the errors calculated from the object detection code. These discrete-time error measurements/estimates, and their correction, can be seen in Figure 6. Since TLE trajectories have position and rate errors, space objects in LEO tend to drift in the finder scope's FoV. The more often the error is estimated and fed to the controller, the less the object drifts in the finder scope's FoV.

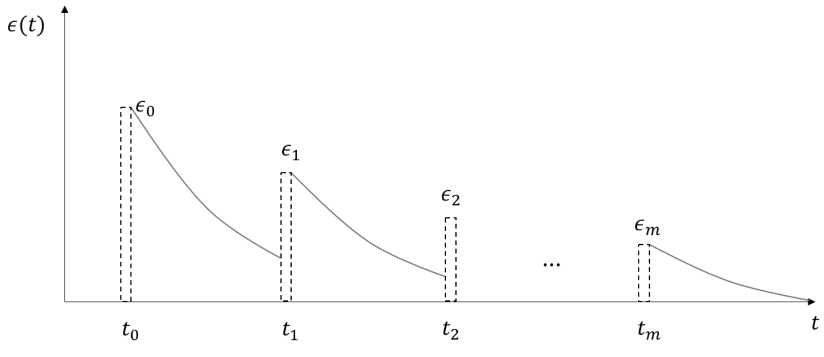

Figure 6: Expected Controller Behavior

3.2. Object Discrimination and Estimation

The object detection code used is designed for cases where the object is visible in the finder scope's FoV with a relatively high SNR. The use of more sophisticated object discrimination algorithms, such as multi-target tracking filters, were considered but not used due to time-constraints.

Before processing the images, the dark frame of the finder scope camera and the estimated background are subtracted to make the images zero-mean. However, performing the background estimation process is computationally inefficient. To perform this more quickly, the background mean is estimated and subtracted, using the following technique.

Let z_i represent the signal that is observed in pixel i of the image. Then, the observed value can be decomposed into the signal from stars and RSOs, background, and noise as s_i, b_i, and w_i, respectively.

$$z_i = s_i + b_i + w_i, \quad w_i \sim \mathcal{N}(0, \sigma) \tag{30}$$

$$\mathbb{E}[z_i] = \mathbb{E}[s_i] + \mathbb{E}[b_i] \tag{31}$$

Computing the mean of both sides using all pixels yields:

$$\sum_i^n \mathbb{E}[z_i] = \sum_i^n \mathbb{E}[s_i] + \sum_i^n \mathbb{E}[b_i] \tag{32}$$

Since most of the image is dark because of the sky, the number of pixels that contain the signal from stars or RSOs are much less compared to the pixels that contain only the background signal. Hence,

$$\sum_i^n \mathbb{E}[s_i] \ll \sum_i^n \mathbb{E}[b_i] \tag{33}$$

$$\implies \sum_i^n \mathbb{E}[z_i] \sim \sum_i^n \mathbb{E}[b_i] \tag{34}$$

Therefore, for any particular image (assuming there is no varying background signal due to the moon or city lights), the mean of the image approximates the background mean well. A simple linear least squares model with a raw image mean predictor variable and a background mean response variable is used. The model predicts the background mean as:

$$f(z_k, \boldsymbol{\gamma}) = \gamma_0 + \gamma_1 z_k \tag{35}$$

where $k = 1, \ldots, m$ denotes the image index. The residual, e_i is found by subtracting the actual background mean b_i from the predicted background mean, $f(i_i, \gamma)$, and is equal to:

$$e_k = b_k - (\gamma_0 + \gamma_1 z_k) \tag{36}$$

The parameters γ_0 and γ_1 are estimated by minimizing the following cost function:

$$\min_{\gamma_0,\gamma_1} \sum_{k=1}^{m} e_k^2 \tag{37}$$

The linear regression model is made with more than 3000 pictures and allowed the background mean to be estimated as a function of the image mean. To build the model, images from different observation sessions are used with a variety of camera gain and integration time settings. As a result, the underlying data is representative of a variety of usable settings and provides a simple and quick approach to estimating the background mean. The maximum residual was roughly 300 counts (16-bit). Given that the intensity count range is [0, 65536], and that the maximum residual is off by 0.45%, the model fits the data well.

Moreover, a thresholding technique was used to construct a binary image based on SNR thresholding. The Scikit-Image toolbox [9] for Python was used to obtain the x,y pixel coordinates of each blob's centroid using a weighted average of pixel intensities in the original image. These centroid coordinates are fed into the Astrometry.net tool, to identify the ra-dec of the objects in the image. The process layout can be see in Figure 7.

Due to noise, most blobs obtained from the images provide no useful information. As such, an intensity based threshold is applied. Then, an area based threshold is applied to eliminate any region in the image that is smaller than the object's expected point spread function (PSF), as seen through the finder scope. The result is a subset of the originally found blobs with useful information.

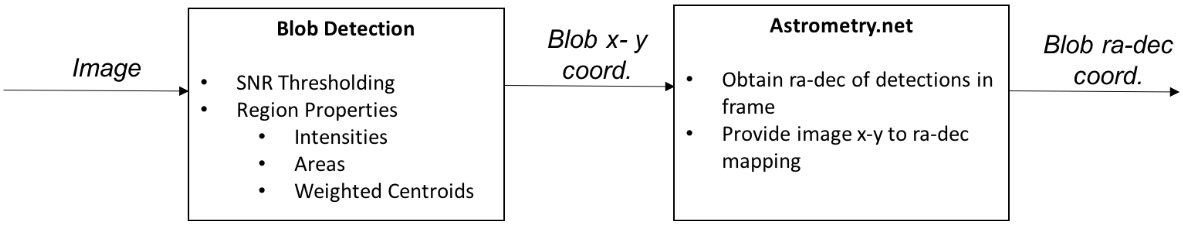

Figure 7: Image Processing Unit

Two techniques can be used to identify the space object within the detection set. One is based on looking at the x-y pixel displacement of all the detections, between adjacent frames, and the other analyzes the ra-dec displacement. To increase the realibility of the identification method, objects that correspond to stars were removed from the image through star subtraction.

For the pixel method, a set of images are considered. In this set, if the detection that moved the least in pixel space is at similar x-y pixel coordinates (by some predefined tolerance), then this is likely the space object of interest. Additionally, if the number of potential matches found is more than 80%, then the most recent object's coordinates are used for the error computation. This is possible since the object's x-y coordinates whilst nominal TLE tracking are expected to be similar. The process is similar to defining Bernoulli trials and counting the successes of the experiment based on the above criteria. When the percentage of success is above 80%, the ra-dec error is computed and fed to the TSX control algorithm.

When comparing two frames, the Hungarian Algorithm [10] is used to associate objects between the two frames. In pixel-space, objects that moved more than a specified Euler distance are eliminated. In ra-dec space, objects that moved more than what is expected from the TLE trajectories are removed. This reduces the frequency of false-positives using the Hungarian Algorithm.

The pixel based method has limitations, particularly when the telescope is correcting estimated errors. Since the criteria of identification requires the object to be static, the object cannot be discriminated when the controller is removing the errors, as its pixel coordinates will change more than the pre-defined tolerance.

The ra-dec method works best when the telescope moves away from the TLE nominal rates, as the object's ra-dec dynamics are independent of the telescope's motion (as opposed to the pixel space method). This method is not affected by the telescope correcting the estimated errors. As such, a Kalman filter is used to provide less noisy feedback to the controller. The object is identified by picking the detection (measurement) with the smallest Mahalanobis distance based on the covariance information from the Kalman filter.

Both of the above methods were implemented and tested for this paper. However, the results shown in this paper are from the pixel based method only. Although these methods are relatively fast in computation time, they are not so robust while tracking objects in different orbit regimes. This can be alleviated by using a multi-target tracking filter such as a Gaussian Mixture-PHD filter or a Generalized Labeled Multi-Bernoulli filter. However, this was outside of the scope of the paper and is future work.

4. Results

Figure 8 shows sample star streak detections, along with the object and TLE position whilst tracking Iridium 914. The green dot represents the expected position from the TLEs. The red dot displays the discriminated objected. The black dots represent the centroids of star streaks. The black box represents the telescope's FoV within the finder scope's FoV. The ra-dec pointing error between the TLE and the telescope's optical axis here are (-0.1552°, -0.2531°). This is different from the TSX's pointing accuracy and requires further investigation. However, the error between the TLE's expected position (green) at that time and the actual position of the object (red) was (-0.1813°, 1.0533°). This provides some experimental insight into the inherent error of a TLE.

Given the computer used, an integration time of 0.5 seconds (2 Hz) is chosen to provide the image processing enough time to complete its tasks and also to capture as much of a space object's signal as possible. At this rate, the image processing code was also faster than the image capturing code. As such, the only lag in the system came from the camera's integration time.

The combined blob detection and Astrometry.net processing frequency was roughly 2.32 - 3.03Hz (180-230 ms and 150-200 ms for detection and Astrometry.net, respectively). This performance was achieved whilst using a Macbook Pro 2015 with a 3.1 GHz Intel i7 processor. The computational times can be reduced further if a desktop computer with newer components is used. The image processing code was tested on such a computer, and the expected image processing frequency was roughly 4-7 Hz.

The results from the computer architecture mentioned above resulted in a number of successful tracking cases. Due to spatial constraints, the trajectory plots and images

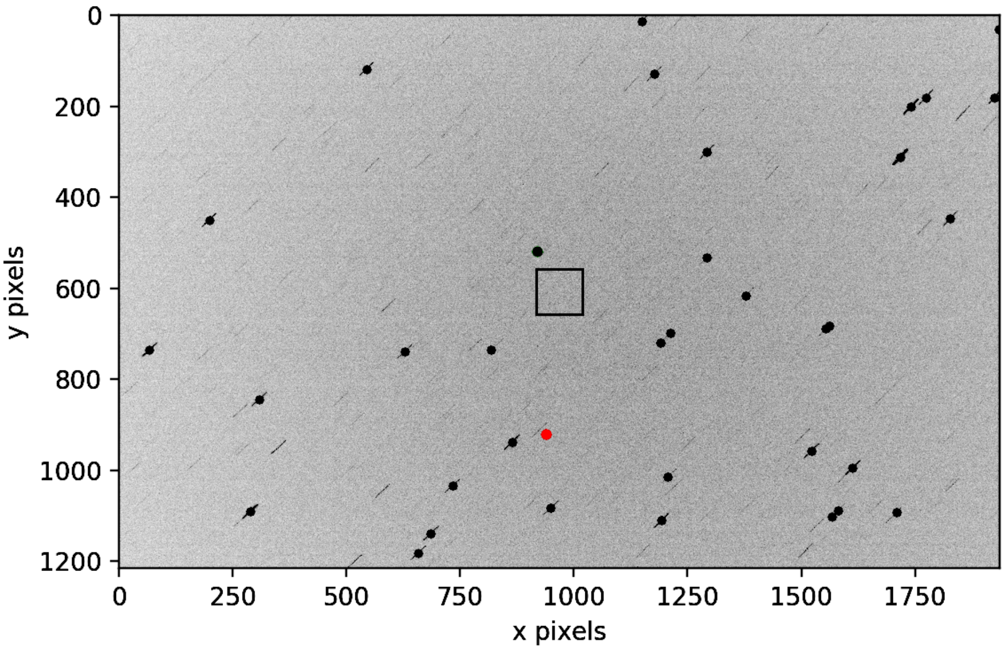

Figure 8: Finder Scope FoV: TLE Position (Green) vs. Object's Measured Position (Red), Iridium 914

of one case are shown. The Iridium 914 satellite, with NORAD catalog ID 24836, was tracked for a duration of just under three minutes. The long-tracking time is as a result of the satellite's trajectory not intersecting the ground station's local meridian.

Figures 9-12 show relevant outputs of the TSX mount (telescope) compared to the TLE trajectories over time. The convergence from an initial pointing offset to the TLE trajectory is visible at the beginning. After converging, the telescope follows the TLE trajectory almost exactly for a brief interval. During this interval, images were taken and the object's ra-dec position was estimated. This corresponds to the large peaks visible in Figures 10-12. The divergence from the TLE's nominal ra-dec trajectories can be seen in Figure 9.

The behavior above can be seen in the ra-dec error plots, in Figure 10. As images were taken and processed, the controller received discrete inputs (error estimates) and corrected for them over time. The initial input is the largest as that is the initial pointing error. After this is corrected, only small deviations away from the telescope's center had to be corrected due to the object drifting.

The effects of Equation 3 can be seen in Figure 12. The nominal trajectory along with the deviations towards the space object's true trajectory can be seen. As the error is corrected in continuous time, the control rates always return to the nominal TLE rates, as mentioned in the previous section.

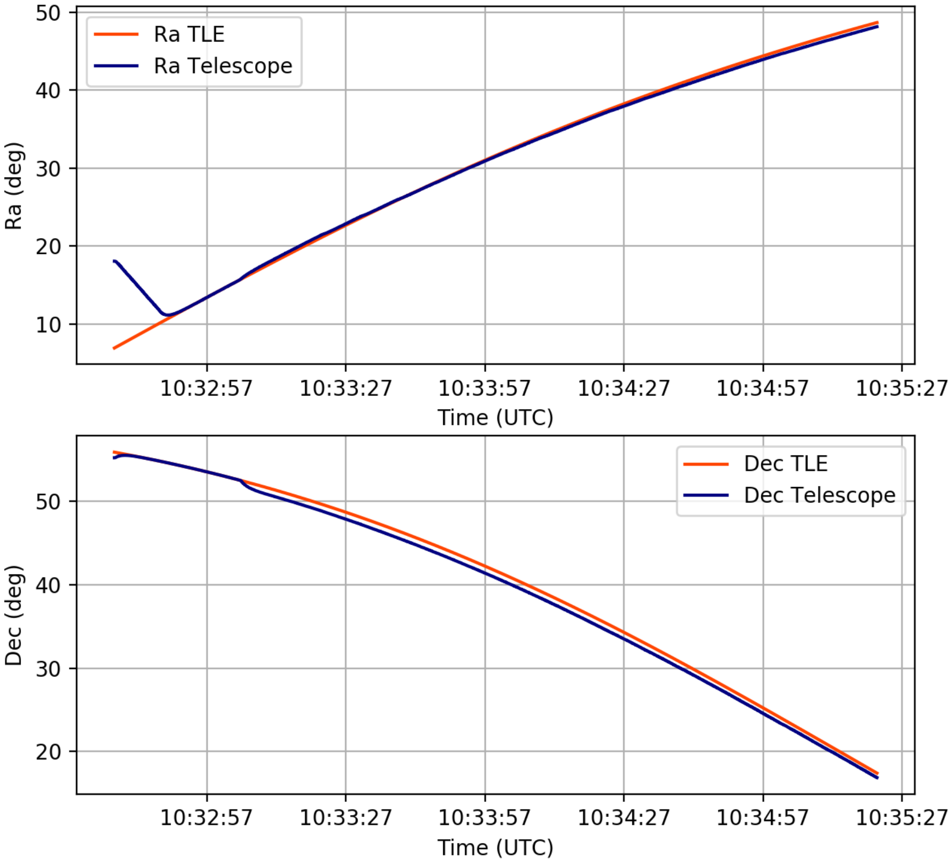

Figure 9: Ra-Dec Telescope - TLE Trajectories

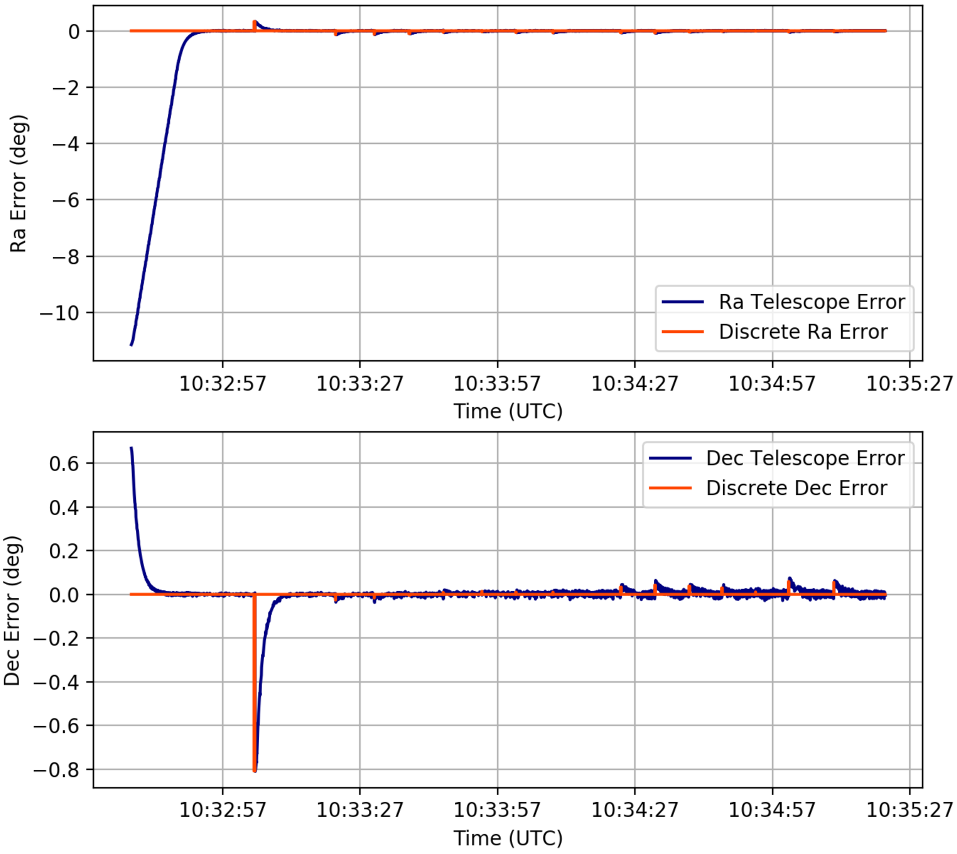

Figure 10: Ra-Dec Error Trajectories

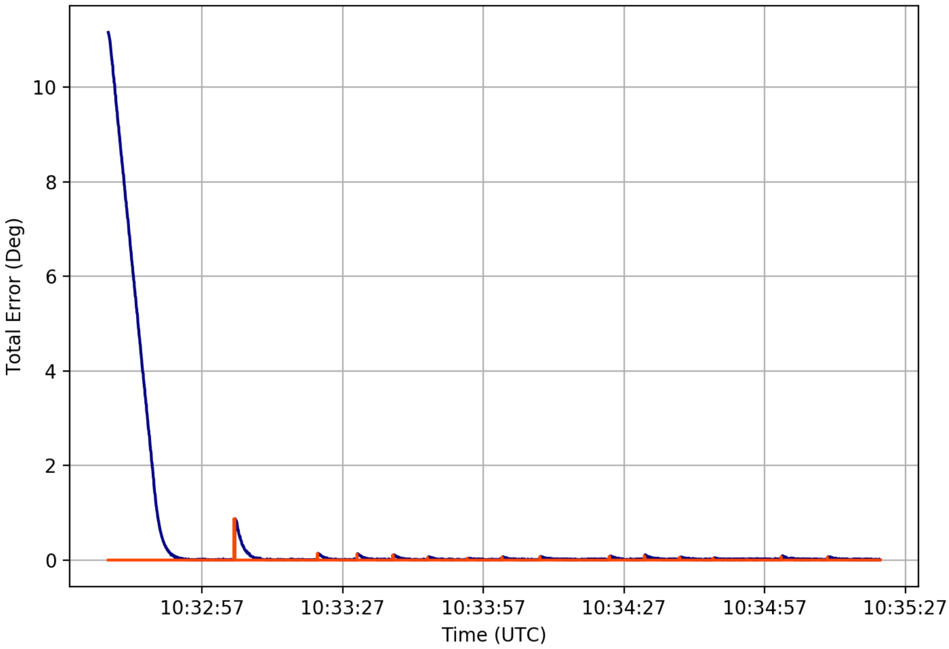

Figure 11: Ra-Dec Error Trajectories

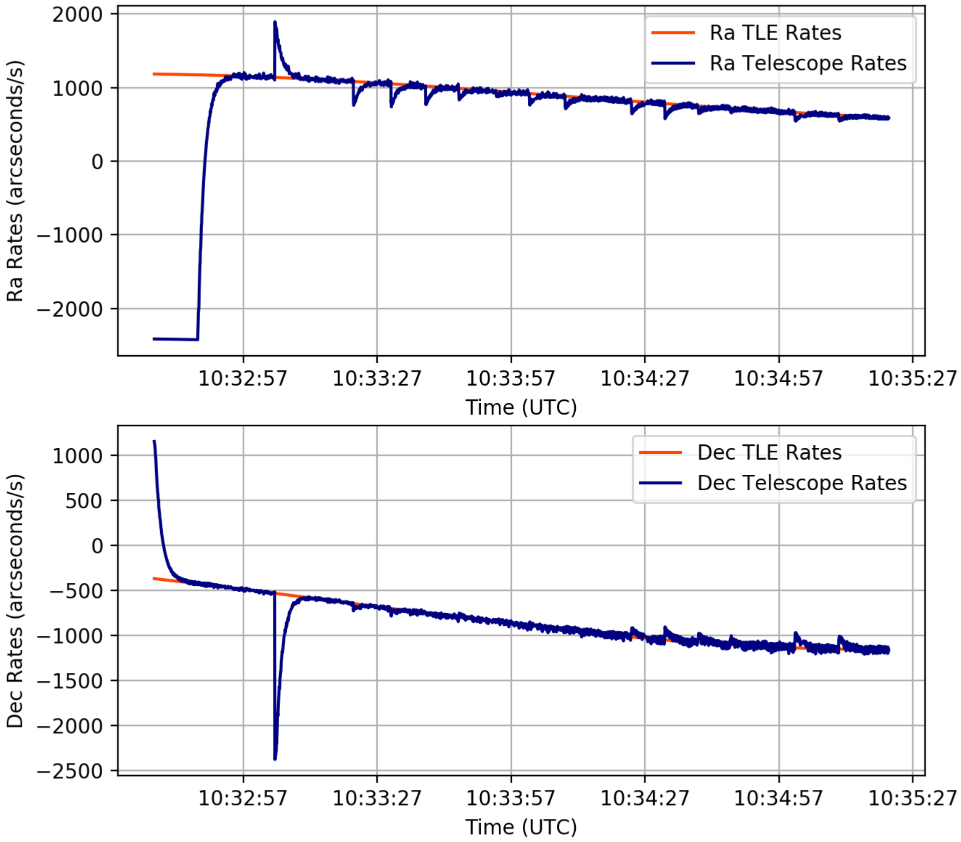

Figure 12: **Ra-Dec Telescope - TLE Rates**

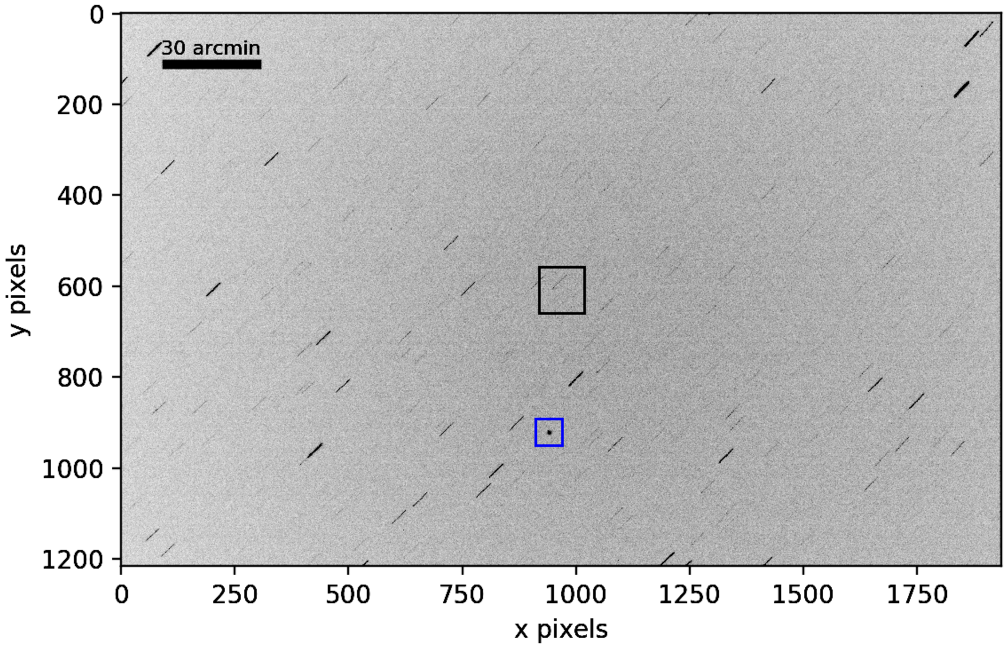

Figure 13: **Before Correction: 10/24/17 10:32:58, Iridium 914, Altitude: 767.3 km. Blue box indicates position of the satellite, whereas black box indicates the main telescope's FoV**

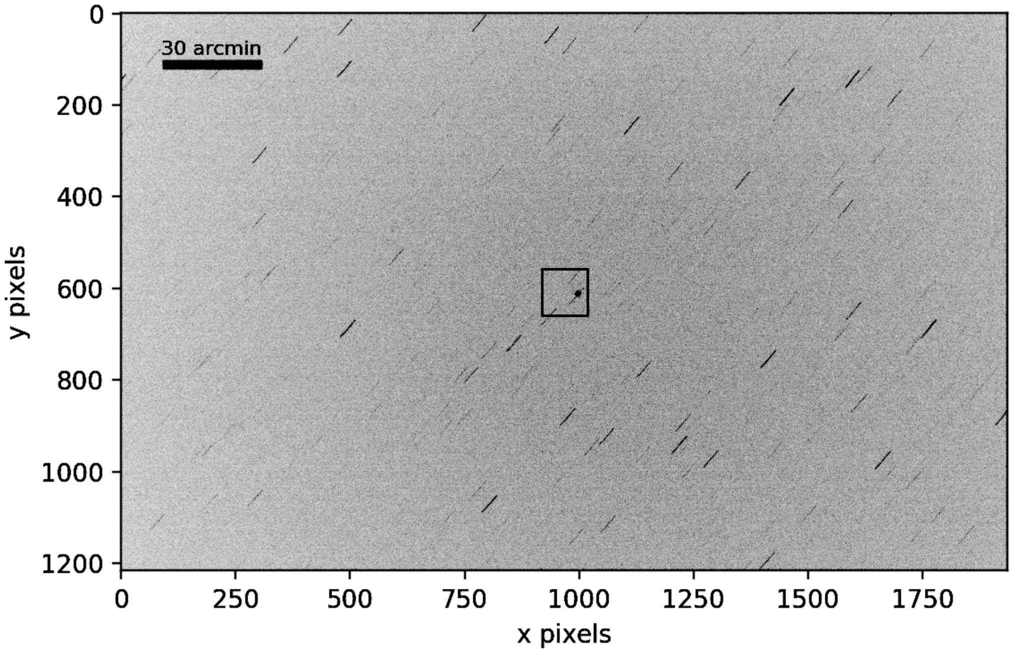

Figure 14: **After Correction: 10/24/17 10:33:39 UTC, Iridium 914, Altitude: 767.3 km. The satellite is within the main telescope's FoV**

5. Conclusion

This project consisted of designing, implementing, and testing a hardware-in-the-loop system to autonomously track space objects with a Raven-class telescope in real-time. This was done using a secondary imaging system, the finder scope, with a wider FoV than GT-SORT. This task was accomplished successfully by interfacing the optical train along with a German-Equatorial mount using Python. Image processing, optimal control and estimation techniques were designed, implemented, and demonstrated on real data.

There are numerous potential extensions to this work. First, a multi-target tracking algorithm can be implemented to robustify the object discrimination process. This would result in smoother trajectories by increasing the measurement update frequency in the controller. Additionally, the image processing from the main telescope camera can be fused with that of the finder scope, resulting in faster convergence and increased accuracy.

Acknowledgments

The authors would like to thank Matthew Harwood for accompanying and helping troubleshoot problems during the recent observation campaigns.

References

[1] M. Mulrooney, P. Hickson, E. G. Stansbery, Orbital debris detection and tracking strategies for the nasa/afrl meter class autonomous telescope (mcat) (2010).

[2] D. Hampf, P. Wagner, W. Riede, Optical technologies for the observation of low earth orbit objects, arXiv preprint arXiv:1501.05736 (2015).

[3] K. J. Abercromby, P. Seitzer, H. M. Rodriguez, E. S. Barker, M. J. Matney, Survey and chase: A new method of observations for the michigan orbital debris survey telescope (modest), Acta Astronautica 65 (2009) 103–111.

[4] F. Alby, M. Boer, B. Deguine, I. Escane, F. Newland, C. Portmann, Status of cnes optical observations of space debris in geostationary orbit, Advances in Space Research 34 (2004) 1143–1149.

[5] D. Liberzon, Calculus of variations and optimal control theory: a concise introduction, Princeton University Press, 2012.

[6] R. F. Stengel, Optimal control and estimation, Courier Corporation, 2012.

[7] J. R. Parker, Algorithms for image processing and computer vision, John Wiley & Sons, 2010.

[8] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches, John Wiley & Sons, 2006.

[9] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, PeerJ 2 (2014) e453.

[10] R. Jonker, T. Volgenant, Improving the hungarian assignment algorithm, Operations Research Letters 5 (1986) 171–175.