# An Automated System to Discover and Track Unknown Geosynchronous Objects Using Ground-based Optical Telescopes

**Yifan Zhou, Lee Devlin, Gemma Cook and Simon Maskell**
*Department of Electrical Engineering and Electronics*
*University of Liverpool*
**Jordi Barr**
*Cyber and Information Systems Division*
*Defence Science and Technology Laboratory, UK*

## ABSTRACT

In this paper, we present a fully automatic system to discover and track unknown objects in geosynchronous orbits (GSO) using an optical telescope. This study is motivated by occurrences where there is limited occupation time on a narrow field of view (FoV) telescope per night and the precise observation time is unknown due to how the telescope is operated. In the method proposed here, the system is initially tasked to survey a wide field of the sky with a fixed number of telescope observations per night. If any GSO object is discovered by the proposed unsupervised machine learning-based detector, the system will start to track the satellite and accordingly schedule future telescope observations. We use particle filters as the tracker which are designed for non-linear problems and will be used to estimate the object states with more sensible uncertainty representations. The system runs in a closed loop and maintains the track of the discovered target with as few telescope tasks as possible.

## 1. INTRODUCTION

A key aspect of space domain awareness (SDA) is the detection, state estimation and propagation of resident space objects (RSOs). This is, more often than not, accomplished via observations taken with electro-optic (EO) sensors (i.e. telescopes). Depending on the observational parameters used, RSOs will form short arcs ('streaks') in ground-based EO images. If the telescope motion is precisely matched to the motion of the target, these arcs can be reduced to an image of the RSO convolved with the telescope's point-spread function (PSF). Searches for uncatalogued RSOs, or those whose state is not well known, are constrained by a sensor's field-of-view (FOV) and the sensitivity, parameters which are generally inversely correlated. A key consideration is therefore how to schedule observations to maximise the opportunity to detect and track RSOs.

Satellites in Geosynchronous orbit (GSO) are responsible for critical tasks such as communication and navigation in both civil and military fields. However, their high orbit keeps them away from atmospheric drag making it difficult to remove them at the end of their operational lifetime. In February 2020 and April 2021, there were two rendezvous and docking activities in this orbit: The Mission Extension Vehicle (MEV-1) with Intelsat-901 and MEV-2 with Intelsat-1002, respectively. The success of these two activities proved that proximity operation of objects in GSO is becoming mature and suggests the need for enhanced situational awareness in these regimes. Both MEV 1 and MEV 2 were monitored using various sensors [1, 2]. These examples demonstrate the significance of discovering and monitoring objects in the GSO regime in order to maintain operational effectiveness.

Existing work on searching for GSOs requires full control of a large FoV telescope, so consecutive images in both time and space can be captured to reduce mis-detections (as in [3, 4, 5, 6]). These approaches mainly spot point detections from a large number of images captured over days rather than considering any orbital information. However, we propose an approach that can be run on any robotic telescope system. In this paper, we consider the Liverpool Telescope where we have low priority in being able to task the telescope and can only submit a number of jobs with ephemerides (right ascension (RA) and declination (DEC)) and time intervals. The telescope will observe the target at a random time in the requested interval depending on the task load on the telescope. Although a smaller time window can ensure more

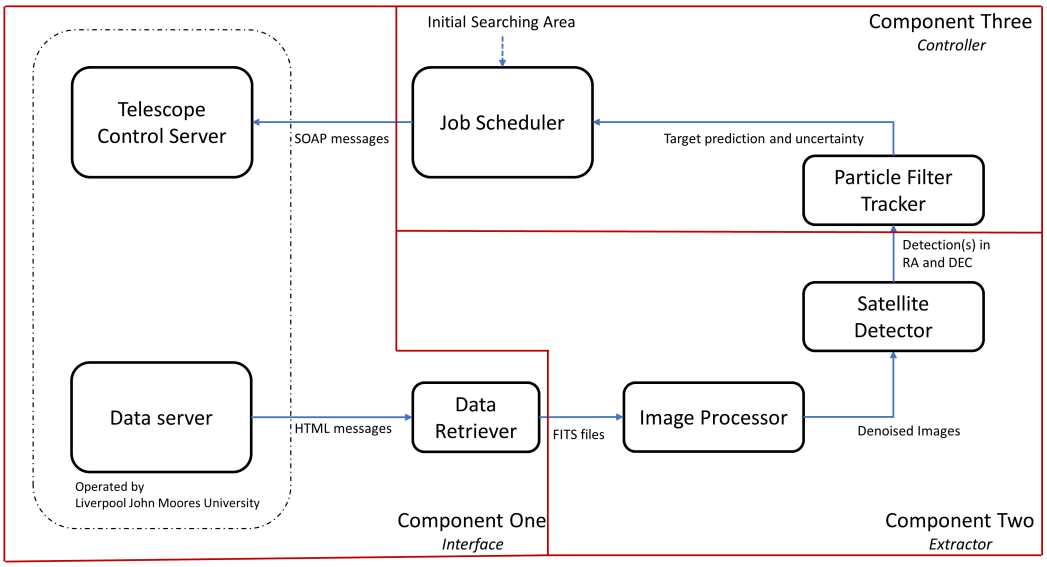



Fig. 1: The architecture of the proposed system.

precise observation times, it increases the chance that the task is skipped. Moreover, we also care about tracking a detected object. We aim to use one observation per night to continuously monitor each object.

This approach can be expanded into a full closed loop system for RSO catalogue maintenance. In order to do so, several autonomous operations are required:

1. A method of determining orbital parameters from images of RSOs,
2. A method of efficiently propagating many orbital states,
3. Sensor management algorithms which assess the value of potential future observation,
4. A control method by which the proposed action can be understood and acted upon by the sensor(s).

The paper is organised as follows. Section 2 will introduce the Liverpool Telescope which motivated our project and the overall architecture of the proposed system. How we extract detections from raw FIT files will be described in section 3. Section 4 will describe the dynamic and measurement models that we used and the basic concept of a particle filter. The filtered states and uncertainties will then be used to schedule the telescope as in section 5. Section 6 includes how the system performs in simulated and real situations using the Liverpool Telescope. Section 7 concludes the paper.

## 2. PROBLEM STATEMENT AND SYSTEM ARCHITECTURE

### 2.1 Liverpool Telescope

The Liverpool Telescope, or 'LT', [7] is a fully robotic telescope owned and operated by the Astrophysics Research Institute of Liverpool John Moores University. For this work we have implemented a back-end program that can submit jobs to the LT as a response to our sensor management system.

The LT is situated at the international Observatorio del Roque de los Muchachos on the summit of the island of La Palma, westernmost of the Canaries, some 200 km off the coast of north-west Africa. The LT is atop an alt-azimuth mount and can be slewed at a maximum rate of 2 °/s. For all collects of geosynchronous targets, the IO:O imager [8] was used which is the main optical-band imaging device and provides a Field-of-View (FoV) of 10 x 10 arcmin.

### 2.2 System Architecture

The proposed system includes three components and each is described as follows (as shown in figure 1).

Component one (*interface*) relies on a control server and data server that have been implemented by the management group of the LT. Our program links to the LT servers to form the management loop.

Component two (*extractor*) is the imaging processor that extracts point detections from LT observations, but also works on the FITS-files from other telescopes. Our proposed algorithm is robust in that it only requires a single image and can work in the presence of a noisy background.

Component three (*controller*) is the back-end algorithm that processes the point detections from component two, initialises and maintains the tracks of objects with a particle filter, and plans the next observations with a three-phase strategy.

## 3. TARGET DETECTION FROM TELESCOPE IMAGES

When objects in GSO have little or no inclination, to an observer on the Earth looking at a fixed azimuth and elevation they appear static. However, the stars in the background will move. Taking a telescope exposure for a number of seconds while observing GSOs and stars will result in points and streaks, respectively. We propose an approach to extract point detections[1] from a single image that can deal with the situations where the point overlays or is close to the streaks. The processing chain of an example image taken by the LT is shown in figure 2.

The first step of the process is to filter out the empty background and identify brighter candidate objects. Due to various conditions (e.g., light, weather, and observation time), the empty areas in the images do not have homogeneous noise properties. A fixed threshold can sometimes produce too many candidates or miss objects. Therefore, we propose a soft threshold that is based on the statistics of the pixel values in the input image.

If we consider the distribution of the pixel values in the image, the pixels that contribute to any object can be seen as outliers to the empty background. The median value of the pixel values represents the fundamental noise level in the image. We use the median rather than the mean because the median is not influenced by extremely bright pixels. The standard deviation of the pixel values is used to filter out outlier pixels which should consist of bright objects. We use a constant value $\alpha$ to control the quantity of the pixels that will be extracted. Greater values of $\alpha$ mean brighter pixels are considered as objects, which increases the reliability of detections. Another constant $c$ is introduced to provide a basic gap between the non-object background and the objects. This is useful to remove false alarms when the distribution of the pixel values is extremely flat (which can be caused by either good astronomical sensing or a noisy background). Thus, we propose this threshold as (1).

$$T^* = median(P) + \alpha \cdot \sigma(P) + c \tag{1}$$

where P denotes the set of all pixel values in the image, $\sigma(.)$ is the standard deviation, and $\alpha$ and c are constant values.

Once the threshold is generated, the input image is converted into a binary map. Basic image morphological operations are done (as suggested in [3, 5]) to connect potential fragmented components. Then, the 8-way connected components are extracted. Each component should reflect position information from at least one object. Each component is examined to detect points.

Consider the situation where one component is contributed by two objects – a star that causes a streak and an object in GEO/GSO orbit which causes a point. The GEO/GSO object is likely to be mis-detected by a threshold-based point detector. However, it is usual that two objects have different brightness. Therefore, we propose to use a Gaussian Mixture Model (GMM) to cluster the pixels based on their positions and pixel values as (2).

$$\begin{aligned} (\mu_i, \Sigma_i) &= GMM(X, 3) \qquad i = 1, 2, 3 \\ X &= [U, V, B] \end{aligned} \tag{2}$$

[1] Objects in Geostationary orbit (GEO) or GSO also produce unresolved observations with a small difference on semi-major and semi-minor axis. We will categorise this case into point detections in this paper.

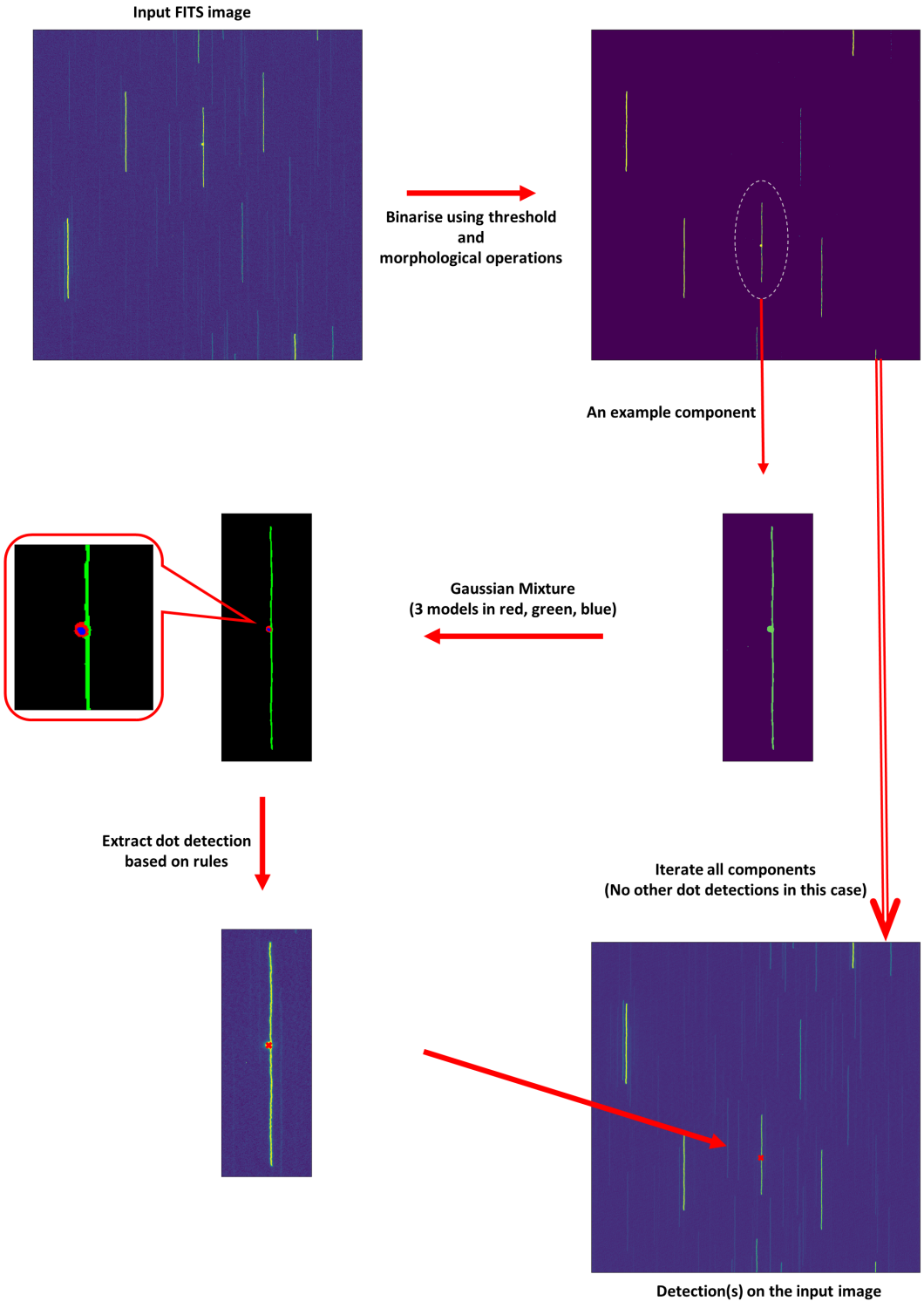
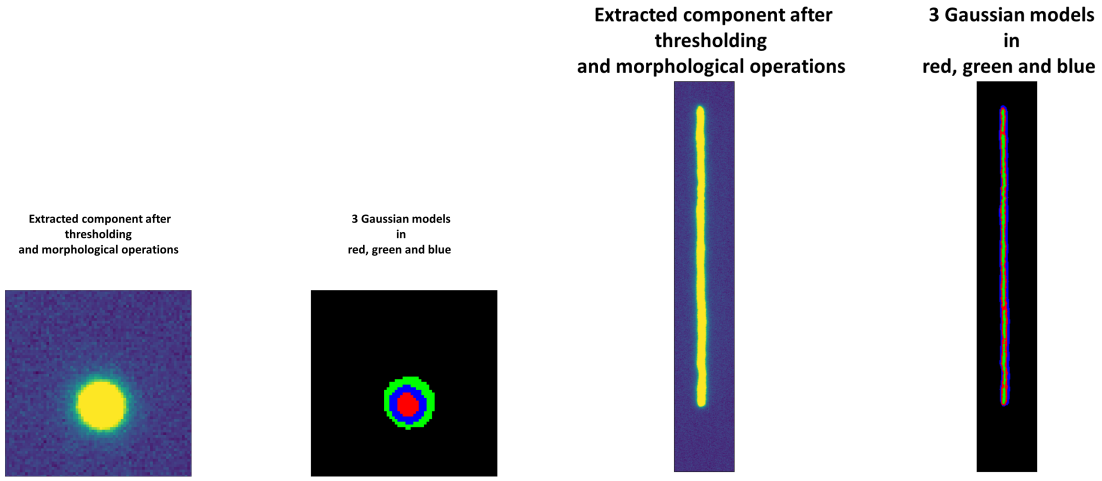



Fig. 2: The processing chain of the proposed GSO object detector.

where $(U,V)$ denotes all the coordinates of the pixels and $B$ denotes all the pixel values; $\mu_i$ and $\Sigma_i$ denote the mean and covariance of model $i$, respectively.

Although the number of clusters can vary, we use a predefined value, which we have selected in this case to be 3, which should extract one cluster for a point, one cluster for a streak, and one final cluster for bright pixels surrounding a streak (see the example in figure 2). Although this looks like an ideal case, it is less likely that one component includes three objects.

In the more common case that the component is attributed to one object, the brightness of the component decreases from the core to the outside smoothly. Therefore 3 Gaussian models with the brightest one in the middle will be extracted. An example of applying a GMM on a component that includes one GSO object is shown in figure 3 (a). Gaussian models are located from the middle to the rounding areas that reflected the different levels of brightness. We can extract the point detection by only considering the middle model (aka. the brightest component). Regarding the streaks that are caused by stars, the Gaussian models still preserves the characteristic of the streak that is long and thin as shown in figure 3 (b). To remove such uninteresting objects, we consider the following detection rules.




(a) An observation of a GSO object.

(b) An observation of a star as a streak.

Fig. 3: Examples of applying Gaussian Mixture model on image components that are contributed by one object.

We treat the Gaussian model, $i_b$, with the smallest brightness mean as the baseline. Candidate detections can only be from the other two Gaussian models and the brightness mean, $\mu_{i_*}(3)$, should be 3 times higher than the baseline, $\mu_{i_b}(3)$. This filters out some false alarms that are unlikely to be of a GSO object. Since we focus on point detections, we also want to limit the shape of the underlying object that is represented by a Gaussian model. It is known that the covariance, $\Sigma$, measures the distribution of points. We can extract the shape of each object from the corresponding element in the covariance matrix $(\sqrt{\Sigma_{i'}(1,1)}, \sqrt{\Sigma_{i'}(2,2)})$. In order to eliminate streaks and keep the points, we propose rules in (3) and (4).

$$\sqrt{\Sigma_{i'}(1,1)}/\sqrt{\Sigma_{i'}(2,2)} \leq 2 \tag{3}$$

$$\sqrt{\Sigma_{i'}(2,2)}/\sqrt{\Sigma_{i'}(1,1)} \leq 2 \tag{4}$$

## 4. TRACKING GEOSYNCHRONOUS SATELLITES

After GSO objects are identified from telescope images, we propose to track them in order to predict their position at a given time. This is useful when we want to monitor the state of an object with the fewest observations, assuming

the object's orbit does not change. As a consequence, the tracker can also detect when the satellite has performed a maneuver. The main focus of the tracker is to estimate the state vector that represents the orbit and it consists of two basic components, a dynamic model and a measurement model. The dynamic model is used to model how the state vector changes over time. The measurement model is used to convert the state vector to a measurement vector e.g. convert the orbit information to right ascension and declination which we can extract from the telescope images (aka. measurements). There are many candidates for a tracker. Since the dynamic and measurement models are non-linear, we propose to use a particle filter which will be described later in this section.

### 4.1 Dynamic model

We elect to use a monopole gravity field to underpin our dynamical model and therefore use Keplerian motion and elements to define orbits. We choose to do this because our focus in this paper is improved handling of the uncertainty rather than modeling the forces upon an orbiting object to high degrees of accuracy. However, we anticipate that improved modeling of the dynamics, e.g. higher-order gravitational effects and non-conservative forces such as radiation pressure, may bring benefit to the approach.

The standard Keplerian elements are not always the most convenient to use, particularly for geostationary orbits. In this situation many of the elements are undefined. For example, in a truly geostationary orbit, the eccentricity $e$ and the inclination $i$ would be zero, which means that both the right-ascension of the ascending node $\Omega$ and the argument of periapsis $\omega$ are undefined. In very near geostationary orbits, small values of eccentricity and inclination can lead to numerical instabilities. This is particularly true when trying to ascertain the uncertainty in these parameters. For these types of orbits, *equinoctial elements*, a reparameterisation of the Keplerian set, has been proposed [9]. This set is defined by six elements: $a$, the semi-major axis; the mean longitude $\lambda_0$, which serves as the measure for where on the orbit the object is on the ellipse; $e_x$ and $e_y$ which are an orthogonal pair of the eccentricity and; $i_x$ and $i_y$, which are similarly for the inclination. The equinoctial elements can be written in terms of the classical orbital elements (where the semi-major axis retains the same definition) as [9]:

$$e_x = e\ \sin(\omega + \Omega) \tag{5}$$
$$e_y = e\ \cos(\omega + \Omega) \tag{6}$$
$$i_x = \tan(i/2)\ \sin(\Omega) \tag{7}$$
$$i_y = \tan(i/2)\ \cos(\Omega) \tag{8}$$
$$\lambda_0 = M_0 + \omega + \Omega \tag{9}$$

Propagating an orbit forwards is performed in a similar way to the standard Keplerian set. First, we find the true longitude $\nu$, which increases linearly with time, from the mean longitude $\lambda_0$. The true longitude is then increased for the time between observations to predict a new position.

### 4.2 Measurement model

The measurement model is a function that translates the state $x_t$ to a set of measurements $y_t$. To do this, we first convert the equinoctial set to a set of Cartesian coordinates in the Earth Centred Inertial (ECI) frame:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{r_{xyz}}{1 + i_x^2 + i_y^2} \begin{bmatrix} (1 + i_x^2 + i_y^2)\cos(\nu) + 2i_x i_y \sin(\nu) \\ 2i_x i_y \cos(\nu) + (1 - i_x^2 + i_y^2)\sin(\nu) \\ 2i_x \sin(\nu) - 2i_y \cos(\nu) \end{bmatrix} \tag{10}$$

where

$$r_{xyz} = \frac{a(1 - e_x^2 - e_y^2)}{1 + e_x \cos(\nu) + e_y \sin(\nu)} \tag{11}$$

We then convert to right ascension and declination in the J2000 frame using standard conversions, e.g. chapter 4 of [10].

### 4.3 Particle filter

We briefly introduce the derivations of the particle filter as follows with pseudo-code presented in Algorithm 1. More detailed descriptions of this technique can be found in [11, 12]. We define the state of the target, $x_t$, in equinoctial elements and the measurement of the target, $y_t$, in RA/DEC at time $t$ as follows.

$$x_t = f(x_{t-1}) + \omega_t \tag{12}$$
$$y_t = h(x_t) + \varepsilon_t \tag{13}$$

where function $f(.)$ is the dynamic model for the target (as described in section 4.1); function $h(.)$ converts target state to the measurement space (as described in section 4.2); $\omega_t$ and $\varepsilon_t$ are the process noise and measurement noise respectively. In the case considered in this paper, we do not have $\omega_t$ and $\varepsilon_t$ is dominated by uncertainty related to converting between the pixel location of the detection in the telescope imagery and the related values of RA and DEC.

Particle filters use the concept of Sequential Importance Sampling (SIS) to numerically approximate estimates of interest using a set of (weighted) particles. The target probability distribution, $\pi(.)$, (which we use to represent the estimation and uncertainty of the states) is then defined on the joint state, $x_{1:t}$, as

$$\pi(x_{1:t}) = p(x_{1:t}, y_{1:t}) \tag{14}$$
$$= p(x_{1:t})p(y_{1:t}|x_{1:t}) \tag{15}$$
$$= p(x_1)\prod_{t'=2}^{t} p(x_{t'}|x_{t'-1})\prod_{t'=1}^{t} p(y_{t'}|x_{t'}) \tag{16}$$
$$= p\left(x_{1:t-1}, y_{1:t-1}\right) p(x_t|x_{t-1})p(y_t|x_t). \tag{17}$$

We also define the proposal that we can draw from to obtain the particles (consider the entire history):

$$q\left(x_{1:t}|y_{1:t}\right) = q\left(x_1|y_1\right)\prod_{t'=2}^{t} q\left(x_{t'}|x_{t'-1}, y_{t'}\right) \tag{18}$$
$$= q\left(x_{1:t-1}|y_{1:t-1}\right) q\left(x_t|x_{t-1}, y_t\right) \tag{19}$$

where $q(x_t|x_{t-1}, y_t)$ is the incremental proposal, used to generate the samples of $x_t$ at each time step.

The weight of each particle can be described as:

$$w_t^i = w_{t-1}^i \frac{p(x_t^i|x_{t-1}^i)p(y_t|x_t^i)}{q(x_t^i|x_{t-1}^i, y_t)} \tag{20}$$

The weights are defined as a mechanism for approximating estimates with respect to a target, and we use the normalised weights that are defined as follows:

$$\tilde{w}_t^i = \frac{w_t^i}{\sum_{i'=1}^{N} w_t^{i'}}. \tag{21}$$

While one could iteratively apply the steps above, over time, the normalised weights will always eventually become skewed: one particle eventually dominates the sum. To address this, resampling is introduced. Resampling replaces the current population of particles with a set with (potentially repeated) indices drawn from the multinomial distribution defined by $\tilde{w}_t^{1:N}$ and equal weights (of $\frac{1}{N}$). Resampling can be seen as quantising the weights, i.e. introducing errors[2]. Since resampling introduces errors, it should only be employed when deemed necessary. This is typically achieved by monitoring the Effective Sample Size (ESS) [13] and only resampling when the ESS drops below a threshold (e.g., $\frac{N}{2}$). The effective sample size is defined as:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}\left(w_t^i\right)^2}. \tag{22}$$

**Algorithm 1** Generic Particle Filter [11]

Define $N_T$ which is the ESS threshold. Starts from $t = 1$.
**for** Each measurement, $y_t$ **do**
  Suppose we have $x_{t-1}^{(i)}$ and $w_{t-1}^{(i)}$ for $i = 1 \dots N$ from the previous time-step.
  **for** Each particle i **do**
    Sample particle from the proposal: $x_t^{(i)} \sim q\left(x_t | x_{t-1}^{(i)}, y_t\right)$.
    Calculate new weights using (20).
  **end for**
  Normalise the weights using (21).
  Calculate the effective sample size, $N_{eff}$, using (22).
  **if** $N_{eff} < N_T$ **then**
    Resample (by drawing the $N$ new particles' indices from the multinomial distribution defined by vector of $w_t^{(i)}$).
    Set all weights to be $\frac{1}{N}$.
  **end if**
  $x_t^{(i)}$ and $w_t^{(i)}$ for $i = 1 \dots N$ are then the new particles.
**end for**

The generic operation of a particle filter is summarised in algorithm 1.

## 5. SCHEDULING THE TELESCOPE

We propose a three-phase approach to discover and track unknown GSO objects given a predefined field of sky.

**Survey Phase**
In the survey phase, the telescope is tasked to look at a field of the sky a number of times per night. Due to the limited FoV, the field of interest consists of a number of windows. Any two windows are ideally not overlaid. A relatively wide time window is predefined so that the telescope will conduct the observations at random points within the window.

**Tracking Phase**
In the tracking phase, the telescope is tasked to look once per night. The target position is given by the tracker (i.e. a particle filter).

**Refining Phase**
In the refining phase, the telescope is tasked to look at a given field of sky once per night. The field of sky under observation is refined by the tracker when the tracker is initialised but the uncertainty is larger than one FoV.

**Transferring between the phases**
The closed loop system will start from the survey phase. Once an object is detected multiple times and enough to confirm a track, it will transfer to the tracking phase to maintain the track of this object. Where a track was lost in the tracking phase, which could be caused by not being able to task the telescope for a long period of time (e.g. telescope under maintenance), the system will transfer to the refining phase. By searching a larger area based on the uncertainty from the tracker, the system will attempt to regain the track. If the track is not recovered, it will be transferred back to the survey phase.

Although the proposed approach includes how to transfer from one phase to the other, the survey phase can be performed all the time. This is when a large field is monitored and there are more than one unknown objects that are supposed to be discovered.

[2]We used a specific resampling algorithm, stratified sampling, in all implementations in this paper (see [11]) since it minimises the size of these errors.

## 6. EXPERIMENTS

Before deploying our approaches onto the real hardware, we generated a simulation to verify the tracker and the scheduling strategy. The simulation is partially aligned with the configurations of LT, but can be modified to simulate other telescopes. We use the simulation to make sure that the real experiment can collect useful information rather than wasting days without discovering any satellite. This simulation will not generate the imagery output from the telescope but RA/DEC observations with Gaussian noise. This section will then report the performance of using the proposed detector from telescope imagery and then present the outcome of using the Liverpool Telescope to discover and track one GSO satellite.

### 6.1 Simulation

As mentioned in section 5, we consider surveying the sky twice per night (00:00-01:00am and 01:00-02:00am) in the survey phase. We consider the location of the telescope as that of the Liverpool Telescope but make the FoV 0.5°. We consider $3 \times 3$ windows which cover the altitude from 42.0° to 43.5° and the azimuth from 220° to 221.5°. Note that altitude and azimuth are both relative to the location of the telescope. We use a Two-Line Element set (TLE) to predict the position of satellites in the sky and, as such, the telescope should be able to observe "USA 170" (whose inclination is 7.5°) at some point.

To generate the telescope measurements for one survey, we first pick observation times in the time window for the 9 observations. We use the *astropy* [14] library to predict the altitude and azimuth at observation time. The observed objects were then recorded. The measurements were gathered per night to mimic the function of the telescope. Note that as the observation times were chosen randomly, the list is not fixed and we choose one to analyse in this section as an example.

In this example, the first two observations (in the survey phase) include one object on day 1 and one object on day 5. We use these two observations to initialise a particle filter and an iterative extended Kalman filter (IEKF, described in [15]) as a comparison. We use both filters to predict the position of the next observation in RA/DEC at a random time from day 6 and 10 to test if the two filters obtain sensible predictions when there were no valid detections over multiple days. The RMSE of the predictions are shown in table 1. It is evident that using a particle filter can obtain better results in predicting future measurements when there are only two historical observations. Both approaches have smaller errors than the telescope FoV (0.5°) which means both algorithms can capture the satellite if it is moved to the tracking phase in this example.

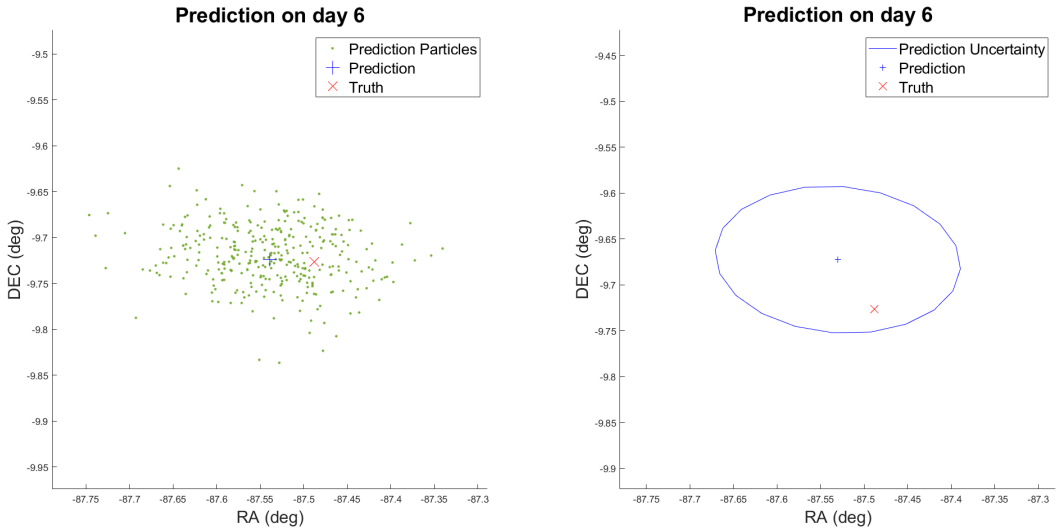
| Method | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 |
|---|---|---|---|---|---|
| Particle Filter | 0.0508 | 0.2660 | 0.0558 | 0.0956 | 0.0695 |
| IEKF | 0.0684 | 0.2821 | 0.0783 | 0.1885 | 0.0656 |

Table 1: The predictive RMSE using particle filter and IEKF.

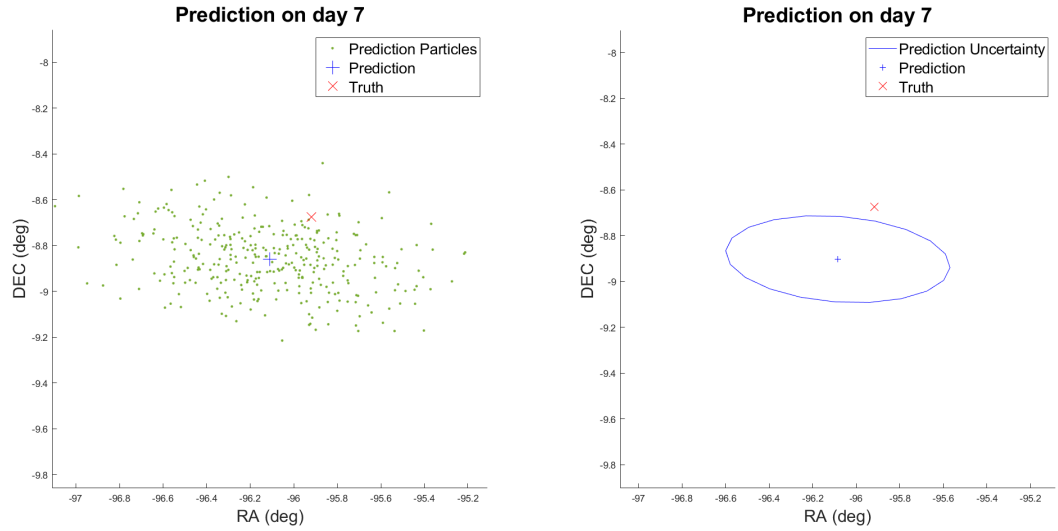
To visualise the uncertainties, we present the predictions and the uncertainties of day 6 and 7 in figure 4. The uncertainty from the particle filter is represented by the particles (green dots). The weights that are associated with the particles ($w_t^i$ in section 4.3) represent their importance. We, therefore, show 25% of the particles with the largest weights. The ellipsoidal uncertainty for IEKF is displayed based on 2-sigma which is 95.4% certainty. As shown in the left figures in figure 4, the truth (red cross) is always within the range of the particles which means the predictive uncertainty is well estimated. However, in the left figures in figure 4 (b) the truth locates out of the ellipse which means the uncertainty of the tracker is underestimated. This could lead to early transferring phases (as described in section 5) when insufficient observations are obtained.

### 6.2 Extracting detections from telescope images

In order to measure the performance of the detection algorithm proposed in section 3, we consider 400 observations we conducted from 64 nights between 23 : 00 and 04 : 00 from October 2021 to July 2022. This is to show how robust the point detector is when we need the system to run for a long time. Some examples are shown in figure 5. The images include: pure background such as figure 5a; poor imagery due to turbulent airflows and complex background such as




(a) Prediction with uncertainty on day 6 by (left) particle filter or (right) IEKF.

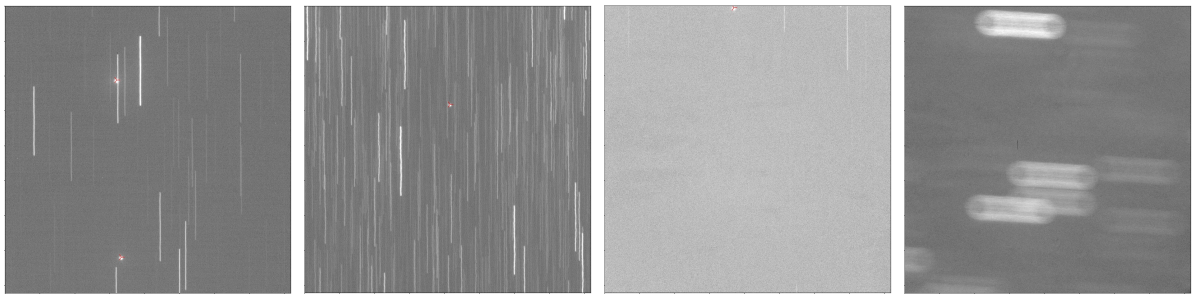



(b) Prediction with uncertainty on day 7 by (left) particle filter or (right) IEKF.

Fig. 4: The predicted measurement and truth on days 6 and 7 using particle filter or IEKF.

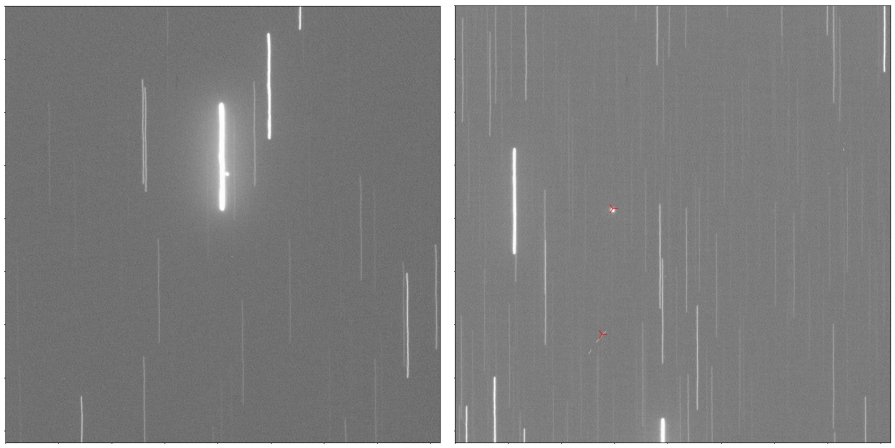
figure 5b; bright and noisy background such as figure 5c; and images that are obviously not usable due to telescope failure figure 5d. However, the proposed algorithm can deal with all these cases without changing the parameters in these examples.

To quantify overall performance, we manually checked all 400 images. There were 275 true detections in total. Among them, the proposed algorithm (using identical parameters) detects 270 of them with 3 false positives. As such, the precision is 98.9% and recall is 98.1%. A mis-detection case and a false positive case are shown in Figure 6. The reason for the mis-detection case is that the streak is too bright while the satellite is small which makes the point source blend into the brighter component. The reason for the false positive example appears to be that an unknown object has given rise to a detection.



(a) Clear background. (b) Complex background. (c) Bright background. (d) Faulty lens.

Fig. 5: Examples of input telescope images and successful detections made by the proposed algorithm in section 3. The detections are illustrated by the red mark.



(a) An example of a miss detection. (b) An example of a false positive.

Fig. 6: Two examples of failed detection by the proposed algorithm in section 3. The detections are illustrated by the red mark.

### 6.3 Object Discovery and Tracking Using Liverpool Telescope

Our experiment was conducted on 13$^{th}$ August 2022 and the telescope was tasked to look in the vicinity of MSG-4 (NORAD ID 40732). This was to make sure that our experiment had an opportunity to track a real-world object. We conducted two surveys per night on the first and second nights. The mosaic image of the first survey is shown in figure 7. Note that the images in each survey were taken at random points in time, so the background varies. All 4 surveys captured the target successfully and a track was initiated by particle filter successfully. After that, the system went into the tracking phase and takes only one image according to the tracker's suggestion.

Figure 8 shows the declination of the predictive track with uncertainty, observations, and telescope FoV. It is obvious that after the two observations on the second night, the uncertainty became smaller than the FoV of the telescope which indicates that we can move from the survey phase to the tracking phase. It can be seen that the observation on the third and fourth nights was within the telescope FoV which matched our observation.

By looking at the left-most part in figure 8, the uncertainty changes from small to very large. That is because the initial particle samples were drawn based on the first observation which was had low uncertainty due to the small measurement noise. However, the partial state vector that is related to the dynamics is highly uncertain. That caused the uncertainty of the prediction to grow quickly which then stopped when the second observation was used to update the tracker.

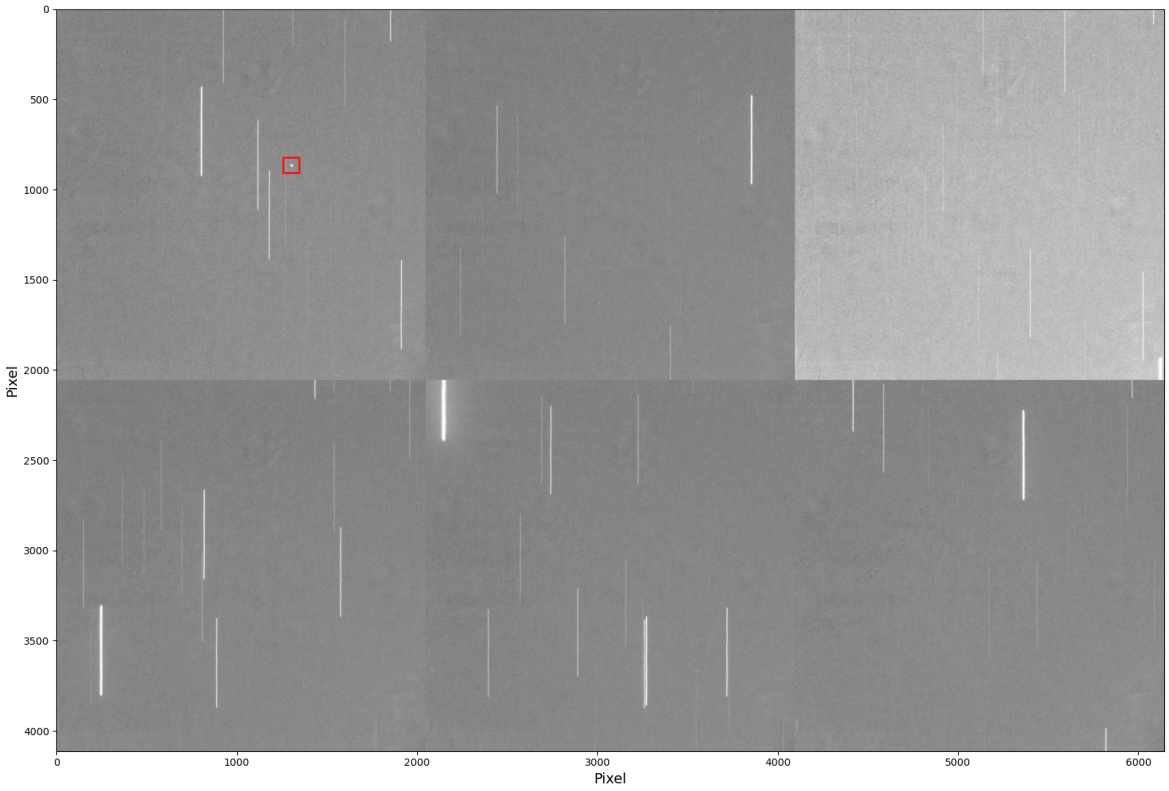



Fig. 7: An example of one survey from Liverpool Telescope. 6 images were taken at random time points in a 1.5 hours time window. Detection is highlighted by the red rectangle.

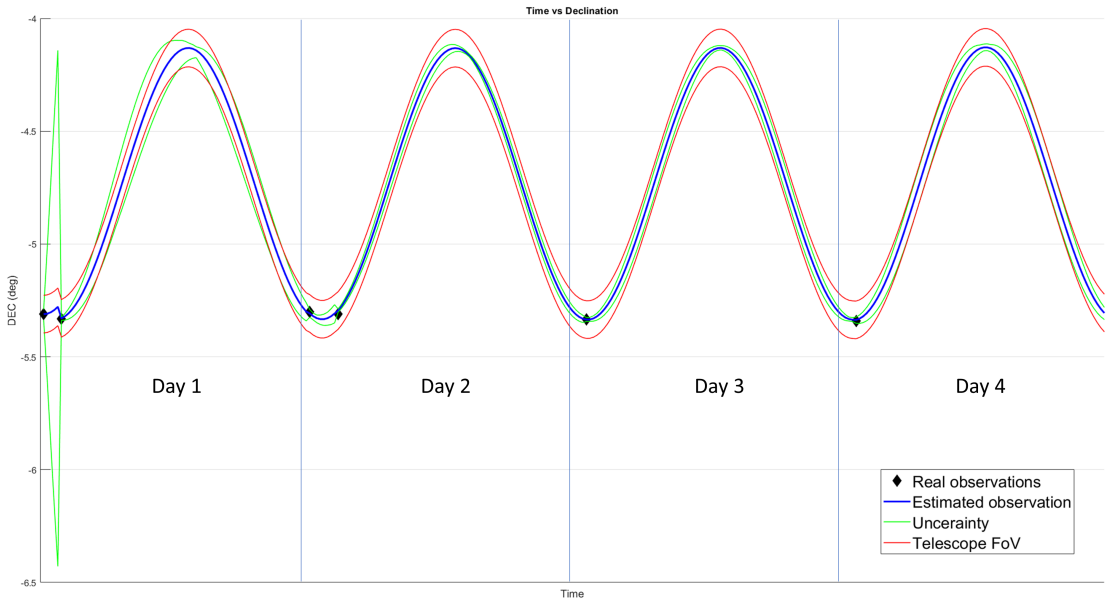



Fig. 8: The predictive track, uncertainty, observations and FoV in declination (DEC) while discovering and tracking a target using the Liverpool Telescope.

## 7. CONCLUSIONS

In this paper, we have presented a closed loop telescope management system that can be used to discover and track unknown geosynchronous objects. The system is designed to work when access to telescope time is limited and the field of regard is much larger than the telescope's FoV. The system does not rely on consecutive observation times and observations can be independent in the search area. Once a target is identified in the survey field a track is initialised. It then aims to take one image per night to maintain the track. The algorithm can also maintain the state estimate and uncertainty when the telescope is not able to work for days such that it can reacquire the target when re-tasked. We have tested the system using the Liverpool Telescope to discover and track the satellite MSG-4 without assuming any prior information regarding the satellite's state.

Further research includes: improving the dynamic model; considering the case of more than one unknown target, which must incorporate a data association method and a multi-target tracker; as well as automatically comparing the confirmed tracks with a catalogue (e.g. of TLEs) to ensure the object is unknown. We are also interested in expanding the spatial and temporal scale of the experiment and hope to discover real unknown objects.

## 8. ACKNOWLEDGMENT

This work is funded in part by the UK Ministry of Defence (MOD) through the *Fusion and Information Theory* (FIT) project, DSTLX-1000143908.

The Liverpool Telescope is operated on the island of La Palma by Liverpool John Moores University in the Spanish Observatorio del Roque de los Muchachos of the Instituto de Astrofisica de Canarias with financial support from the UK Science and Technology Facilities Council.

## REFERENCES

[1] Travis Bessell, James Frith, Lauchie Scott, Jovan Skuljan, Roberto Furfaro, and Vishnu Reddy. Phantom echoes: A five-eyes sda experiment to examine geo rendezvous and proximity operations simon george, andrew ash defence science & technology laboratory, uk. 2020.

[2] Simon George, Alexander Agathanggelou, Grant Privett, Philippa Halpin, William Feline, Andrew Ash, Paul Chote, Lauchie Scott, Jovan Skuljan, Jason Alvino, et al. Phantom echoes 2: A five-eyes sda experiment on geo proximity operations. In *Proc 22nd AMOS Conf*, 2021.

[3] Myrtille Laas-Bourez, Gwendoline Blanchet, Michel Boër, Etienne Ducrotté, and Alain Klotz. A new algorithm for optical observations of space debris with the tarot telescopes. *Advances in Space Research*, 44(11):1270–1278, 2009.

[4] Sandrine J Thomas, Jeffrey Barr, Shawn Callahan, Andy W Clements, Felipe Daruich, Juan Fabrega, Patrick Ingraham, William Gressler, Freddy Munoz, Doug Neill, et al. Vera c. rubin observatory: telescope and site status. In *Ground-based and Airborne Telescopes VIII*, volume 11445, pages 68–82. SPIE, 2020.

[5] James A Blake, Paul Chote, Don Pollacco, William Feline, Grant Privett, Andrew Ash, Stuart Eves, Arthur Greenwood, Nick Harwood, Thomas R Marsh, et al. Debriswatch i: A survey of faint geosynchronous debris. *Advances in Space Research*, 67(1):360–370, 2021.

[6] Hao Luo, Jing-Hui Zheng, Wei Wang, Jian-Jun Cao, Jie Zhu, Guo-Ping Chen, Yong-Shuai Zhang, Chang-Shun Liu, and Yin-Dun Mao. Focusgeo ii. a telescope with imaging mode based on image overlay for debris at geosynchronous earth orbit. *Advances in Space Research*, 69(6):2618–2628, 2022.

[7] The Liverpool Telescope. `https://telescope.livjm.ac.uk/`.

[8] The Liverpool Telescope. `https://telescope.livjm.ac.uk/TelInst/Inst/IO0/`.

[9] R. A. Broucke and P. J. Cefola. On the equinoctial orbit elements. *Celestial Mechanics*, 5(3), 1972.

[10] D. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 2013.

[11] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[12] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

[13] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[14] Adrian M Price-Whelan, BM Sipőcz, HM Günther, PL Lim, SM Crawford, S Conseil, DL Shupe, MW Craig, N Dencheva, A Ginsburg, et al. The astropy project: building an open-science project and status of the v2. 0 core package. *The Astronomical Journal*, 156(3):123, 2018.

[15] Bradley M Bell and Frederick W Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.