MULTI-OBJECTIVE MONTE CARLO TREE SEARCH FOR AUTONOMOUS SPACE SURVEILLANCE WITH UTILITY VECTOR OPTIMISATION

Phuong Linh Ngo, Steve Gehly, Marco Langbroek, and Pieter Visser

Delft University of Technology, Kluyverweg 1 2629 HS, Delft, The Netherlands,
Email: {P.L.N.Ngo, S.Gehly, M.Langbroek-2, P.N.A.M.Visser}@tudelft.nl

ABSTRACT

The multi-objective sensor tasking problem is framed as a Partially Observable Markov Decision Process (POMDP). To solve the POMDP, recent work has explored the use of Monte Carlo Tree Search (MCTS). While MCTS is traditionally applied to single-objective problems, this work extends its use to multi-objective sensor tasking. The approach optimises three objectives: discovering unregistered space objects, estimating their orbits with limited measurements, and refining catalogued orbits through tracking. Instead of scalarising multi-objective returns, the algorithm maintains a utility vector, driving MCTS toward Pareto-optimal solutions. Macro- and micro-action levels are distinguished to manage the complexity of the decision problem. At the micro level, initial implementations focus on stripe scanning for search, refinement of the admissible region for follow-up observations, and an information-theoretic selection strategy for tracking. While the current implementation is limited and exploratory, it provides a proof-of-concept and identifies key directions for further development.

Keywords: Sensor tasking, Monte Carlo Tree Search, Decision problem, Multi-objectives, Pareto opimisation.

1. INTRODUCTION

To support sustainable practices in space and ensure secure operational activities, there is a growing interest in autonomous methods for space surveillance. Space surveillance seeks to provide comprehensive insights into space objects, including their origins, trajectories, mission goals, physical attributes, and rotational dynamics. Within the context of sensor management for space situational awareness (SSA), these requirements introduce competing objectives. The goal of sensor management is to optimise the utilisation of sensor capacities to simultaneously achieve all SSA objectives. This paper focuses on the following research question:

How can a single sensor system be tasked to efficiently meet multiple objectives in SSA?

For surveying the space environment, optical sensors can be tasked with scanning stripes to search for and re-observe objects in geosynchronous orbit (GEO)[10, 18, 5]. The distribution of most GEO objects forms a ring defining the Laplace plane. To cover the entire distribution, it is sufficient to place the stripe at a fixed right ascension angle containing the ring and scan this declination stripe continuously in a leakproof manner. Scanning multiple stripes instead of just one allows for (re-)observations at different positions along the orbit, refining orbit determination.

Schubert et al. [16] support stripe scanning as an effective method for systematically discovering new objects. Furthermore, they propose extending the objective space to improve initial orbit determination (IOD) by optimising a greedy sensor pointing reward function. Similarly, Frueh et al. [6] frame sensor tasking as an optimisation problem, aiming to maximise the observed objective area while satisfying specific criteria to enhance detection rates compared to classical stripe scanning methods.

Previous research on sensor tasking has primarily focused on single-step approaches, maximising immediate rewards and producing optimal yet myopic solutions. Fedeler [4] applies Monte Carlo Tree Search (MCTS) to optical sensor systems to find near-optimal solutions over the long term. Unlike previous methodologies, MCTS employs a multi-step approach, identifying a sequence of pointing directions while considering the future behaviour and evolution of the state and action space.

Ngo et al.[13] extends MCTS to a bi-objective problem, balancing searching and tracking. However, this long-term optimisation focuses on balancing sensor time resources rather than the value each observation adds to maintaining a space object catalogue. Existing literature explores the use of information theory to quantify the information gained from observations, guiding the optimisation of target-tracking strategies. Information gain has been described using Fisher information, Rényi divergence, and Kullback-Leibler divergence [21, 7, 12].

Despite these advancements, there remains a gap in addressing sensor tasking as a *multi-objective* and *multi-step* problem. A *multi-step* approach seeks an optimal tasking strategy over the long term, while a *multi-*

objective perspective incorporates the need to gather as much information as possible from the space environment to meet disparate needs of users, the SSA community and space operators. The proposed approach considers three main objectives: searching for unregistered space objects, scheduling follow-up measurements to facilitate and confirm the objects' existence to add them to the catalog, and refining orbit information for catalogued objects through continuous tracking. This research aims to MCTS to perform multi-objective optimisation, preserving the rewards of each objective in a utility vector rather than scalarising multi-objective returns. The MCTS algorithm prioritises pointing solutions based on dominance in the utility space, iteratively refining the decision tree to favour promising solution paths. The final selection is near Pareto-optimal, improving the balance across multiple SSA objectives. At the current stage, the implementation is limited and exploratory, but nevertheless provides a proof-of-concept and identifies key directions for further development.

Furthermore, this research introduces a redefined action space, structured as a tuple of macro and micro actions. Micro actions are tactical, immediate choices linked to high-level user objectives, represented as macro actions. This distinction enables balancing short-term, objective-driven decisions within the micro action space with long-term strategic planning across multiple objectives in the macro action space. The searching objective is achieved using a stripe scanning algorithm. The tracking micro action prioritises candidates based on their potential to maximise information gain about catalogued objects over the observation campaign. The follow-up objective is realised by computing the constrained admissible region (CAR) [1] from newly registered detections, with the reward function favouring strategies that minimise uncertainty regarding the CAR.

The structure of this paper is as follows: Sec. 2 provides background on partially observable Markov decision processes (POMDPs), Pareto optimisation, and MCTS. Sec. 3 details the implementation of MCTS in an SSA environment. Finally, Sec. 4 summarises key findings and offers recommendations for future research.

2. BACKGROUND

2.1. Partially Observable Markov Decision Problem

To address the research question, the sensor tasking problem is framed as a Partially Observable Markov Decision Problem (POMDP). The term *partially observable* reflects the sensor's limited knowledge of the environment, requiring a probabilistic belief in the state [15]. A POMDP is defined by the septuple $\langle S, T, A, O, H, R, \gamma \rangle$:

- S represents the multi-target state space, encapsulating knowledge of the space environment, maintained through an observation campaign,
- T denotes the conditional transition probabilities between states,
- A is the action space, representing decisions made by an agent (e.g., an optical sensor),
- O is the observation space, where states and transitions are only partially observable,
- H represents the measurement function that incorporates uncertainty, linking observations to the underlying state,
- R quantifies the reward behind the associated change in the state and action,
- $\gamma \in [0, 1)$ is the discount factor, prioritising immediate rewards when optimising the sensor tasking strategy. In other words, a discount factor less than 1 decreases the importance of future rewards and values immediate rewards more [3].

2.2. Monte Carlo Tree Search

To solve the POMDP, this work explores Monte Carlo Tree Search (MCTS), a reinforcement learning algorithm that dynamically builds and evaluates a decision tree to optimize task selection for an agent, such as an optical sensor. MCTS seeks near-optimal long-term solutions by identifying a sequence of pointing directions while accounting for state space evolution.

Hayes et al. [9] propose an expectimax search tree using MCTS, where expectimax refers to maximising the expected utility of a decision sequence. Here, utility models feedback for multiple objectives, while reward represents single-objective outcomes. The expectimax decision tree consists of decision and chance nodes: a decision node represents the environment and indicates that a decision (further referred to as an action) has to be made. In contrast, a chance node is a child of a decision node and denotes the outcome resulting from the selected action. The root node is always a decision node. To derive the optimal sensor pointing strategy over the entire observation campaign, the accumulated expected rewards of each branch are summarised in a utility vector and propagated backwards to the root of the tree. The utilities are updated and built up over a series of iterations, during which the algorithm develops experience in deciding which actions should be taken at each time to meet the multiple objectives within the observation campaign.
MCTS therefore constructs a partial version of the full decision tree, which is biased to focus on paths more likely to lead to higher utility. This allows the algorithm to achieve near-optimal outcomes while remaining computationally tractable.

MCTS operates in two stages: planning and execution. The planning stage consists of four phases:

- **Selection:** The algorithm traverses the tree, choosing child nodes based on a balance of exploration

(discovering new options) and exploitation (selecting known high-utility paths). Random selection can introduce inefficiencies, whereas guided selection improves decision quality.

- **Expansion:** A new action is chosen based on a predefined policy, leading to the generation of a new decision node, as shown in Fig. 1.
- **Simulation (also Roll-out):** A predefined policy rolls out from the expanded node until a termination state, such as the end of the observation campaign, is reached.
- **Backpropagation:** The resulting cumulative utility vector propagates back through the tree, updating utility values and visit counts, which influence future selections.

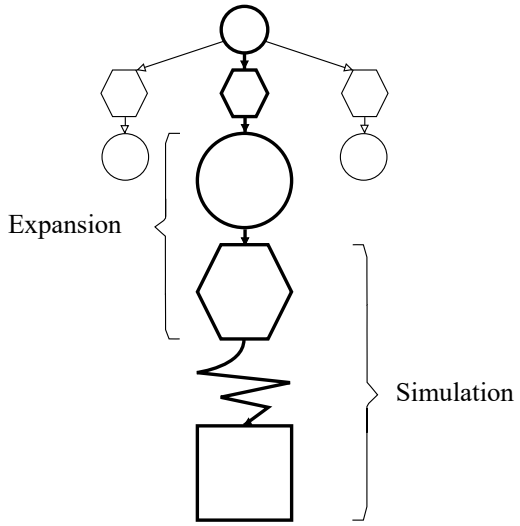

Figure 1. Decision tree with a newly expanded and simulated branch. Decision and chance nodes are represented by circular and hexagonal shapes, respectively. The quadratic node illustrates the termination state of the observation campaign.

2.3. Basic formulation of multi-objective decision-making

Multi-objective decision-making algorithms seek to return a set of parameters that best optimise the following problem defined in its general form:

$$\begin{aligned} \min / \max\{y\} = f(\bar{x}) &= [f_1(\bar{x}), f_2(\bar{x}), ..., f_m(\bar{x})] \\ \text{subject to: } \bar{x} &= (x_1, x_2, ..., x_n) \in \boldsymbol{X} \\ \bar{y} &= (y_1, y_2, ..., y_m) \in \boldsymbol{Y} \end{aligned} \tag{1}$$

Here, $\bar{x}$ represents an n-dimensional decision vector, and $\bar{y}$ is an m-dimensional objective vector. The compromise over different (conflicting) objectives in $f(\bar{x})$ leads to a set of potential trade-off solutions, also called Pareto-optimal solutions.

A solution is Pareto-optimal if it is not dominated by any other solution, meaning no objective can be improved without degrading another. Given two solutions $\boldsymbol{x^{(1)}}$ and $\boldsymbol{x^{(2)}}$, $\boldsymbol{x^{(1)}}$ dominates $\boldsymbol{x^{(2)}}$ if:

1. $\boldsymbol{x^{(1)}}$ is no worse than $\boldsymbol{x^{(2)}}$ in all objectives, i.e. $f_j(\boldsymbol{x^{(1)}}) \not\triangleright f_j(\boldsymbol{x^{(2)}})$ for all $j = 1, 2, ..., M$
2. $\boldsymbol{x^{(1)}}$ is strictly better than $\boldsymbol{x^{(2)}}$ in at least one objective, i.e. $\exists k \in \{1, 2, ..., M\}$, $f_k(\boldsymbol{x^{(1)}}) \triangleleft f_k(\boldsymbol{x^{(2)}})$

The operator $\triangleleft$ accounts for both minimisation and maximisation and denotes that (1) is better than (2) as $f_k(\boldsymbol{x^{(1)}}) \triangleleft f_k(\boldsymbol{x^{(2)}})$. By iteratively comparing all solutions, non-dominated solutions form the Pareto-optimal set.

3. MONTE CARLO TREE SEARCH FOR SENSOR TASKING IN SPACE SITUATIONAL AWARENESS

Building on the general methodology of MCTS described in the previous section, this part introduces a novel approach to constructing the decision tree for optimising long-term pointing strategies. The objective is to maximise the expected utility of a pointing strategy while addressing three key goals:

1. Searching for unknown space objects,
2. Conducting follow-up observations to refine initial orbit determination, and
3. Tracking specific targets of interest.

These objectives form the so-called *macro action space*. In the following study, the size of the macro action space is hence restricted to three objectives. From this point forward, the terms *macro action* and *objective* will be used interchangeably.
Since the tree starts with a decision node, it is first necessary to define the role of this node in the context of the SSA sensor tasking problem. Hence, Sec. 3.1 introduces the characterisation of the nodes in the decision tree distinguishing between chance and decision nodes. The following Secs. 3.2 - 3.4 explore each macro action in detail, beginning with a methodological overview, followed by exemplary settings and test cases.

3.1. Node characteristics

As noted in Sec. 2.2 a decision node encodes the current environment based on which a decision has to be made. For the given sensor tasking problem, this environment represents the status of the SSA catalogue that maps the evolution of the multi-target state space. Each decision node stores the following parameters:

- The accumulated time resources spent on previous search tasks,
- The propagated environment under the influence of the previous IOD actions,
- The propagated environment under the influence of the previous tracking actions,
- The current pointing direction of the sensor

Since the root node does not yet have any child nodes, the expansion phase is entered, and a new chance node needs to be generated. The child node is sampled from the macro action space, determining whether the underlying node will perform a search, IOD, or tracking task. Based on the sampled macro action, the corresponding pointing task is selected from the micro action space, which contains all potential pointing directions. The micro action space varies depending on the sampled objective (see Sec. 3.2, 3.3 and 3.4). The selection of a micro action follows a predefined reward function for the macro action space. At this level, the choice is made using a greedy policy, ensuring that the highest-reward micro action is prioritised. The selected micro action then leads to the creation of a chance node, which stores the following parameters:

- the sampled macro action
- the derived micro action
- the execution time of the underlying sensor task

The newly created chance node is directly followed by the next decision node, which represents the updated environment after incorporating the effects of the chance node.

Independent of its type (decision or chance), a node additionally stores the following characteristics:

- the current utility value,
- the current utility vector and
- the number of visits to the current node.

The utility value is a scalar that records the number of solutions that dominate the current node. The utility vector, on the other hand, contains the individual rewards accumulated from each micro action space.

3.2. Search

Once the algorithm determines that the current node should be extended through a searching task, it invokes a stripe scanning method introduced by Herzog et al. [10], Flohrer et al. [5] and Siminski [18]. In this method, a stripe composed of multiple declination fields is generated and placed at a fixed right ascension angle. To optimise visibility and maximise the signal-to-noise ratio, the stripe is positioned near the Earth's shadow boundary.

The scanning process begins with the sensor positioned at the lowest declination field within the stripe. The sensor captures images while gradually moving upwards, covering the entire stripe. To prevent objects from escaping detection between consecutive scans, the stripe length (i.e., the number of declination fields) is adjusted in order to guarantee leakproofness. This adjustment accounts for the angular velocity of target objects, the sensor's field of view (FOV), and its slewing velocity, as given by:

$$N_{decField} = \left\lfloor \frac{t_{leakproof} + t_1 - t_2}{N_{exp}} \right\rfloor \tag{2}$$

with $t_{leakproof}$ as the time duration it takes for a space object to pass through the sensor's FOV, t_1 as the longer of either the repositioning time to the next declination field or the sensor's readout time, t_2 as the longer of either the repositioning time from the last to the first declination field or the sensor's readout time, and N_{exp} as the number of images taken in one declination field. Under the assumption that the space object takes one day to orbit Earth, the detection algorithm is adapted to geosynchronous orbits.

Fig. 2 provides a schematic overview of the scanning process. Based on its slewing velocity, the sensor repositions to the lowest declination field of the scanning stripe. It then settles at the new pointing location and prepares for the upcoming measurements. Fig. 2 also shows a zoomed-in illustration of the scan process within one declination field: In total, the sensor takes N_{exp} images of the same declination field before repositioning to the next one. After each image exposure of duration t_{exp} the sensor requires t_{read} to readout the data. The duration of scanning one declination field is given by Eq. 3.

$$t_{decField} = N_{exp} \cdot t_{exp} + (N_{exp} - 1) \cdot t_{read} \tag{3}$$

3.2.1. Example of Stripe Scanning Method

To demonstrate the performance of the stripe scanning method, consider an optical sensor located at a geodetic latitude $\phi = 6°$, longitude $\lambda = -37°$ and at altitude $h = 0\,\mathrm{m}$. Furthermore, the sensor has the following technical specifications: exposure duration $t_{exp} = 8\,\mathrm{s}$, settling time $t_{set} = 7\,\mathrm{s}$, readout time $t_{read} = 7\,\mathrm{s}$ and an elevation cutoff angle of $\epsilon_{cut} = 5°$. To maximise the apparent brightness of GEO objects, the scanning stripe is positioned near the Earth's shadow, centered at a geocentric right ascension of $\alpha_{\oplus} \approx 192.24°$ on the night of March 24, 2025, at 22:01:02.62 UTC. Fig. 3 illustrates the scanning stripe configuration. The sensor's parameters enable a leakproof scan across $N_{decField} = 6$ declination fields, with each field being exposed $N_{exp} = 5$

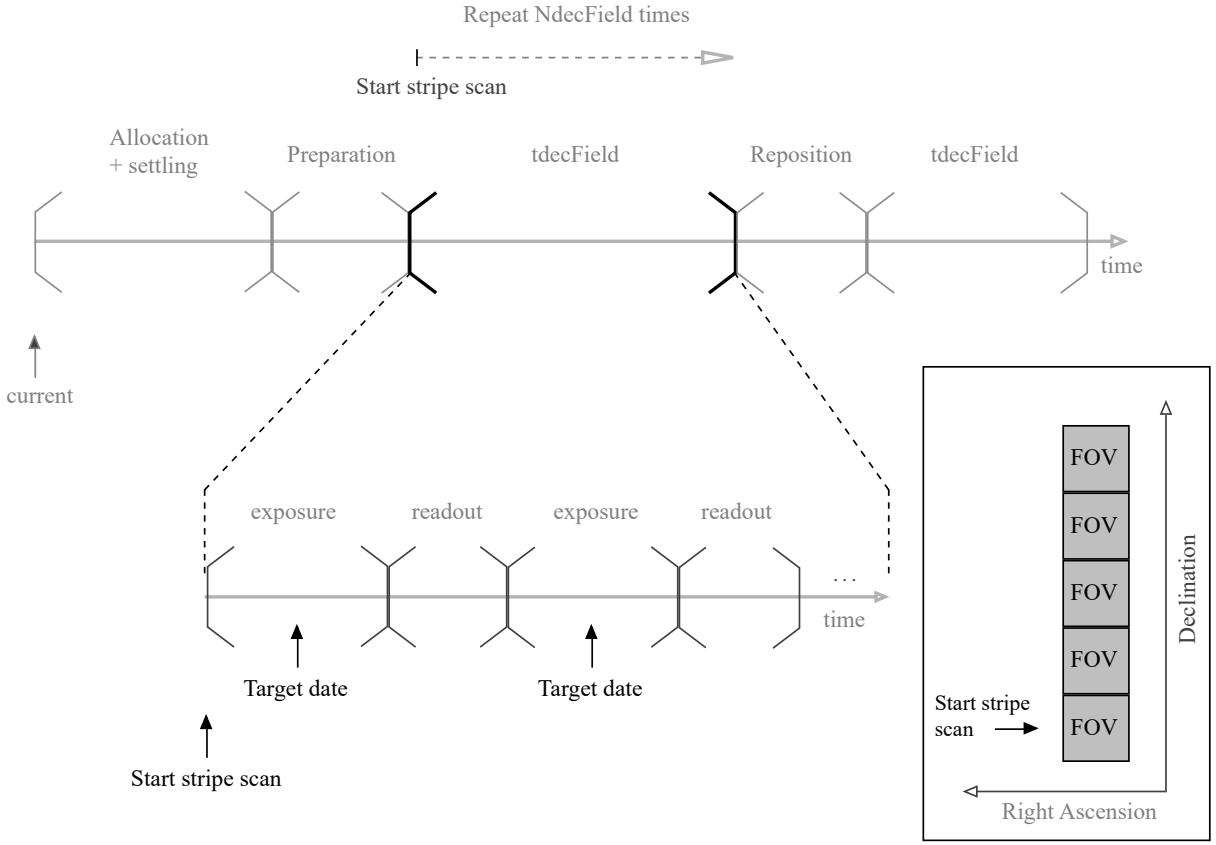

Figure 2. Operational timeline of search task. The box at the right side illustrates a scanning stripe with five declination fields.

times. The number of declination fields is determined using Eq. 2, where $t_1 = 9\,\mathrm{s}$ is the repositioning time to the next declination field and $t_2 = 7\,\mathrm{s}$ set to the readout time. Note that the stripe is not centered around the equatorial but around the Laplace plane. The entire scan of the stripe takes approximately $475\,\mathrm{s}$.

The performance of the stripe scanning method is evaluated based on the number of detected space objects. For this assessment, the entire GEO catalogue, consisting of 1,029 space objects, was retrieved from Spacetrack [19]. The dataset is filtered to include only objects with a mean motion of $n \in [0.99; 1.01]$ and an eccentricity $\epsilon < 0.01$. The Two Line Elements (TLE) are propagated to the measurement epochs of the stripe scan tasks. A detection is recorded whenever a propagated object enters the sensor's field of view (FOV) at a given measurement epoch, assuming a detection probability of one.

As Fig. 3 shows, a total of 23 detections were recorded, which corresponds to eight unique detections. Despite setting the number of exposures to five, certain objects were not captured in all five images. This was the case for all detections within FOV5. Hence, all of them do not reveal at least three angular measurement pairs, which are necessary to perform an initial orbit determination. To further evaluate the efficacy of the collected measurements in deriving an orbit of the other objects, the detections associated with object 22724 were selected for an angles-only IOD using Gooding's method [8]. Unfortunately, the resulting initial state estimate is quite poor. This is primarily due to the minimal spread between the measurement pairs, coupled with the influence of measurement noise. With a time interval of just $15\,\mathrm{s}$ between two measurements, the three angular pairs used for the IOD covered less than $0.04\,\%$ of the actual orbit.

These findings highlight the need for improvements in orbit determination accuracy for objects detected using the stripe scanning method. One potential enhancement involves introducing an additional macro-action for re-observation, which is discussed further in Sec. 3.3.

3.3. Initial Orbit Determination

In the case that the algorithm decides to extend the current node with an Initial Orbit Determination task, the angular measurements that were recorded in a previous search task are used to construct an attributable. For optical observations this attributable is presented by a vector that contains two angles and two angular rates:

$$\mathscr{A} = \left[\alpha, \delta, \dot{\alpha}, \dot{\delta}\right]^T$$

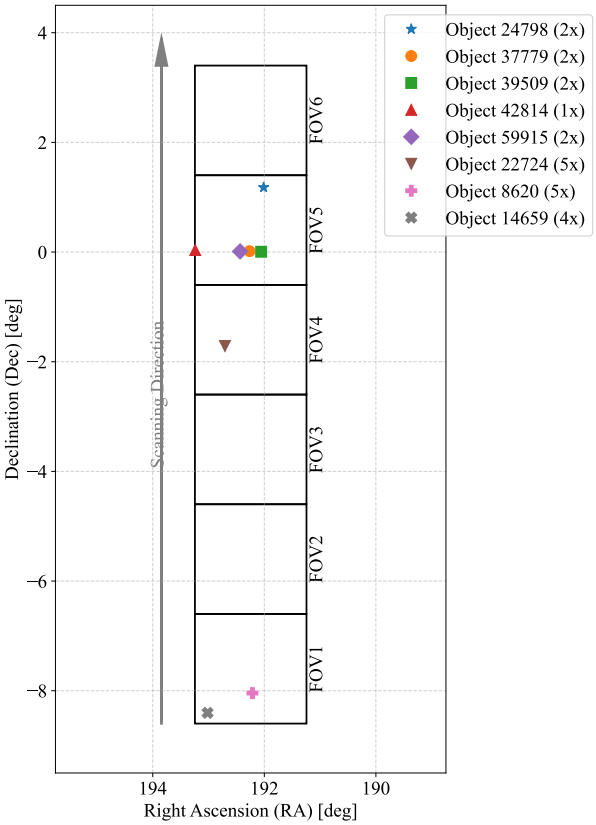

Figure 3. Simulation of a stripe scan placed at the geocentric right ascension $\alpha_\oplus \approx 192.24°$ and centered at the Laplace plane. The scan takes approximately 475 s and captures eight space objects. The number of detections for each object is found in brackets in the legend. Throughout the stripe scan the space objects are moving eastwards (i.e. to the left).

where α is the topocentric right ascension, δ the declination, and the dot denotes the time rate of change of the topocentric angles. Derived from the observation geometry and two-body energy, a two-dimensional space of admissible solutions called the *Admissible Region* (AR) is defined. For optical observations, this region is spanned by the range and range-rate dimension, and its unconstrained form contains all solutions of the under-determined orbit system [14]. The set of solutions can be further constrained by incorporating apriori information about the semi-major axis and eccentricity of the orbit, resulting in the so-called *Constrained Admissible Region* (CAR).

This study builds upon the work of DeMars and Jah [1] who derived the CAR and approximated it using a Gaussian Mixture Model (GMM). An unscented Kalman Filter for Gaussian Mixtures (GMUKF) is then employed to propagate the AR to a designated target date where a potential measurement is used to update and further reduce the set of orbital solutions. The updated GMM and the predicted GMM (without measurement update) are each merged and pruned into a single Gaussian distribution [20]. In that way the information gain of an IOD task can be quantified by a measure like the Kullback-Leibler (KL) divergence [11]. Its closed form is given by Eq. 8

$$\begin{aligned} d_{KL}\left(\mathcal{N}_1(\mu_1,\,\Sigma_1)||\mathcal{N}_2(\mu_2,\Sigma_2)\right) = \\ \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) \\ + \left(\mu_2-\mu_1\right)^T\Sigma_2^{-1}\left(\mu_2-\mu_1\right) - n\Bigg] \end{aligned} \tag{4}$$

Maximising KL divergence (i.e. information gained) defines the reward function within the IOD micro action space. This means a myopic greedy algorithm is employed to identify the most suitable candidate whose Admissible Region should be further refined through a new measurement.

3.3.1. Example of IOD micro action tasking

To demonstrate the performance of the IOD micro action function, an exemplary object from Sec. 3.2.1 is reused to generate a CAR represented by a GMM. As shown in Fig. 3, object 22724 has five detections, corresponding to five angular pairs associated with this object. To utilize all available observations, a linear regression is performed on all measurements to derive the angular rates. The choice for a linear polynomial is justified since the tracklet is very short [14]. For object 22724, the resulting attributable is

$$\begin{aligned} \mathscr{A}_{22724} &= \left[\alpha, \delta, \dot{\alpha}, \dot{\delta}\right]^T \\ &= \begin{bmatrix} 192.83° \\ -1.74° \\ 0.0041\,°\,\mathrm{s}^{-1} \\ -0.0010\,°\,\mathrm{s}^{-1} \end{bmatrix} \end{aligned}$$

on March 24, 2025, at 22:05:40.00 UTC.
The AR shall be constrained by semi-major axes ranging from $a \in [40.000\,\mathrm{km}; 45.000\,\mathrm{km}]$ and eccentricities $\epsilon \leq 0.02$, since the object is assumed to orbit in GEO. The derived CAR is depicted in Fig 4 and it can be seen that it is primarily constrained by the eccentricity limit.

To facilitate efficient processing (e.g., state propagation, filtering), the CAR is approximated by a Gaussian Mixture Model (GMM). Since the CAR is derived from a very short tracklet, information on which range and range-rate combination are more likely than any other is sparse. This justifies the approximation of the CAR as a uniform distribution in the form of a GMM. Additional constraints are applied to further simplify the construction of the GMM:

1. All Gaussian components in the mixture have equal weights.
2. The means are evenly distributed over the AR approximation.

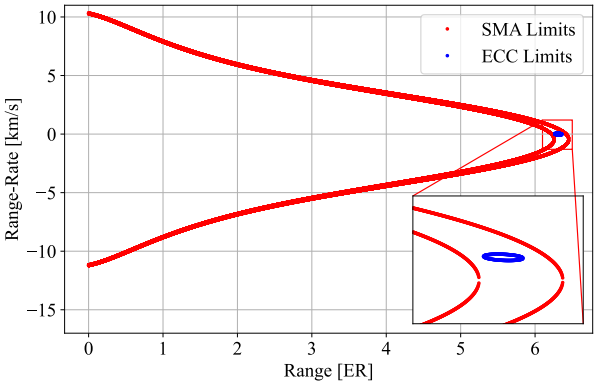

Figure 4. Constrained Admissible region of object 22724 from its tracklet generated by the stripe scanning method.

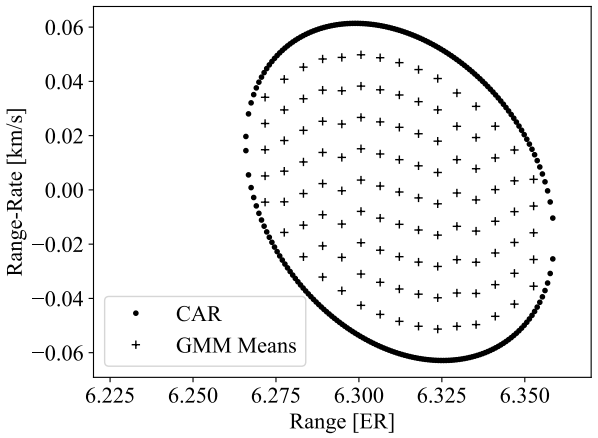

Figure 5. Location of Gaussian Mixture means in CAR of object 22724 on March 24, 2025, at 22:05:40.00 UTC derived from the angular measurements of the initial detections.

3. The GMM is assumed to be homoscedastic, meaning that all components share a common standard deviation σ.

Fig. 5 shows the locations of the Gaussian mixture means within the CAR of object 22724. For further details on constructing a GMM approximation, readers may refer to DeMars and Jah [1]. Tab. 1 summarises key parameters (such as the maximum standard deviation σ_{max}, the number of GM components L and the optimal standard deviation $\tilde{\sigma}$) necessary to approximate the CAR as a GMM.

A subsequent follow-up measurement update is scheduled $1\,\mathrm{min}$ after the original declination stripe scan to accommodate sensor slewing. In this example, let the observed geocentric angular position of the satellite be given by

$$\alpha_{\oplus,22724} \approx 193.56^\circ \qquad \delta_{\oplus,22724} \approx -1.87^\circ$$

Each GMM component shown in Fig. 5 is propagated under Keplerian dynamics through an Unscented Kalman Filter (UKF) towards the measurement update epoch T_{update}. The settings for the unscented transform and generation of the χ - points are listed in Tab. 2 The updated GMM at T_{update} is then specified by the updated Gaussian probability density functions (PDF) of each individual mixture component:

Table 1. Specific parameters to recover the CAR approximation as GMM for object 22724.

Parameter	Values
$\sigma_{\rho,max}$	$500\,\mathrm{m}$
$\sigma_{\dot{\rho},max}$	$10\,\mathrm{m\,s^{-1}}$
L_ρ	15
$L_{\dot{\rho}}$	5
$\tilde{\sigma}_\rho$	$0.0456\,\mathrm{m}$
$\tilde{\sigma}_{\dot{\rho}}$	$0.1531\,\mathrm{m\,s^{-1}}$

Table 2. Setting of Unscented Transform

Description	Parameter	Values
Spread of χ points	α	1
Prior knowledge of state distribution	β	2
Scaling factor	κ	-3
Tailedness of distribution	$kurt$	3
State dimension	n	6

$$p\left(\boldsymbol{x}_k|\boldsymbol{Y^k}\right) = \sum_{l=1}^{L} \omega_{l,k}^{+} p_g\left(\boldsymbol{x}_k; \boldsymbol{m}_{l,k}^{+}, \boldsymbol{P}_{l,k}^{+}\right) \tag{5}$$

where k denotes the time variable and in this case refers to T_{update}, L is the number of GM components, and $p_g\left(\boldsymbol{x}_k; \boldsymbol{m}_{l,k}^{+}, \boldsymbol{P}_{l,k}^{+}\right)$ are the updated GM components . The new weights $\omega_{l,k}^{+}$ are retrieved by means of the Gaussian multivariate measurement likelihood $\zeta_{l,k}$ such that

$$\omega_{l,k}^{+} = \frac{\zeta_{l,k}\omega_{l,k}^{-}}{\sum_{j=1}^{L} \zeta_{j,k}\omega_{j,k}^{-}} \tag{6}$$

To measure how much the posterior distribution given by Eq. 5 differs from the conditional GMM $p\left(\boldsymbol{x}_k|\boldsymbol{Y^{k-1}}\right)$ without measurement update at time T_{update}, the KL divergence is computed. However, for GMMs, a closed-form expression of the KL divergence does not exist, and the usual approach is to approximate this measure [2]. In the given example, the conditional and posterior GMMs are each pruned and merged into two single Gaussian distributions. Herefore, a pruning threshold $T = 1 \times 10^{-3}$ and a merging threshold $U = 1 \times 10^{9}$ are selected. A mixture component whose weight falls below T is discarded from the distribution. The remaining components

are pairwise compared to compute the Mahalanobis distance. A pair, whose Mahalanobis distance falls below U gets merged together. Since the merging threshold U is very large in this example ($U = 1 \times 10^9$), all remaining components merge into a single Gaussian component with a total weight of one. This effectively simplifies the KL divergence calculation to a comparison between two single Gaussian distributions.

3.4. Track

Once the algorithm decides to extend the current node with a tracking task, a myopic greedy algorithm identifies a suitable tracking candidate through the following steps:

Step 1: Target Date Initialisation

First, the future angular position of each candidate is computed at a *target date*, a future instant at which tracking would begin. This date is offset from the current time instance T by a duration t_{shift}:

$$t_{shift} = t_{alloc,def} + t_{set} + t_{prep} + 0.5t_{exp} + t_{step}$$
$$T = T + t_{shift} \tag{7}$$

where $t_{alloc,def}$ denotes a pre-defined time duration to reposition the sensor to a new direction, t_{set}, t_{prep} and t_{exp} account for the settling, preparation and exposure time, respectively, and t_{step} is the incremental evaluation step duration.

Step 2: Visibility Check

In time increments of duration t_{step}, the algorithm assesses whether a tracking task for the candidate at hand is feasible. The algorithm evaluates each candidate for up to 200 s. If visibility conditions become unfavorable or the sensor cannot reposition quickly enough, the candidate is removed from further consideration. The visibility conditions at the target date are assessed through several criteria:

- Is the candidate in the sensor's field of regard?
- Is the candidate eclipsed by Earth's shadow?
- Is the angular separation between the Sun and the candidate with respect to the observer at least 60°?
- Is the angular separation between the Moon and the candidate with respect to the observer at least 20°?

If a candidate fails any of these visibility criteria, it is excluded from consideration.

Step 3: Feasibility Check (Slewing Duration)

In the next step, the actual slewing duration t_{alloc} required to relocate the sensor from its current pointing direction toward the candidate is computed. If $t_{alloc} > t_{alloc,def}$ or if any visibility check fails, the candidate is marked as unsuitable at the current target date. In such cases, the step size t_{step} is incremented by 1 s and Eq. 7 is updated accordingly. The process repeats until a suitable target date is found.

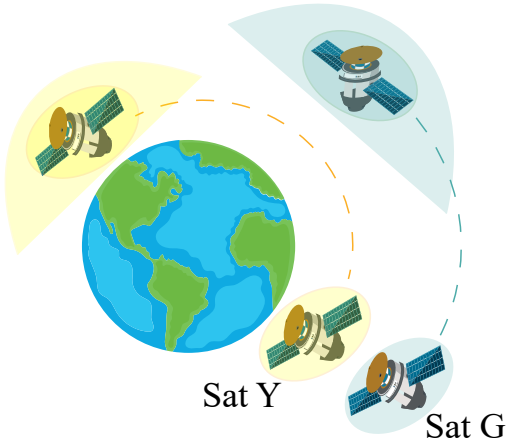

Figure 6. Illustration of two tracking candidates (Sat G and Y) whose initial state uncertainty is propagated in time. The posterior PDF (oval shape) is compared with the conditional PDF (banana shape) to measure the information gained from a measurement update.

Step 4: Simulated State Update and Information Gain

Once a valid target date is identified, the candidate undergoes a simulated state update. Its angular position is converted into a right ascension/declination measurement pair relative to the topocentric inertial frame. This angular pair is used to simulate an update of the candidate's state and state covariance estimate using an Extended Kalman Filter (EKF) [17]. The EKF returns the predicted state uncertainty as well as the updated state uncertainty. The KL divergence, which quantifies the reduction in uncertainty and thus the information gain from the tracking task, is calculated to compare the probability distributions before and after the update. Assuming Gaussian distributions, it is given by Eq. 8 [22, 11].

$$d_{KL}\left(\mathcal{N}_1(\mu_1, \Sigma_1)||\mathcal{N}_2(\mu_2, \Sigma_2)\right) = \frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) - n\Bigg] \tag{8}$$

Fig. 6 illustrates an example of two tracking candidates . It can be seen that a measurement update shows a greater uncertainty reduction of Sat G compared to Sat Y.

Step 5: Utility Evaluation

The algorithm then assigns a tracking utility vector

$\boldsymbol{u}_{track,i}$ for each candidate i:

$$\boldsymbol{u}_{track,i} = \begin{bmatrix} d_{KL,i} \\ \Delta T_i \end{bmatrix} \tag{9}$$

where $d_{KL,i}$ represents the KL divergence as information gain, while ΔT_i is the waiting time between the sensor's readiness and the upcoming target date to track the candidate as derived in Eq. 7.

Step 6: Micro Action Selection

To optimise candidate selection, the algorithm prioritises higher information gain and shorter waiting times. The candidate with the lowest number of dominating solutions with respect to $\boldsymbol{u}_{track}$ is selected as the tracking target. In case multiple solutions reveal the same dominance, the final target among these candidates is selected based on a random policy.

As detailed in Sec. 2.2, MCTS not only exploits the strategy that yields the most promising myopic results but also explores alternative strategies by expanding multiple nodes. To avoid redundant expansions, tracking candidates already selected by sibling nodes are excluded from further consideration.

3.4.1. Example of multi-target tracking

To evaluate the performance of the tracking micro-action function, the MCTS is initialised with a single-objective task that exclusively focuses on tracking. In this example scenario, the algorithm aims to maintain accurate orbital information for three targets. Following the optimisation strategy outlined in Sec. 3.4, the algorithm generates a set of instructions specifying which targets to track, at what times, and in which sensor directions. The tracking candidates are presented in Tab 3.

Following the steps in Sec. 3.4, this example applies tracking candidate selection to the TDRS satellites, expanding the MCTS with a new chance and decision node.

Step 1: With the sensor settings presented in the example of Sec. 3.2, the current time instance T =2025-03-24 T22:01:02.620 UTC is shifted forward by $t_{shift} = 20\,\mathrm{s}$. Hereby, $6\,\mathrm{s}$ are reserved to prepare the sensor for the upcoming tracking task and $2\,\mathrm{s}$ are reserved to allocate the sensor to its new pointing position. The incremental step size t_{step} is set to $1\,\mathrm{s}$. Accordingly, the current target date becomes T22:01:22.620 UTC.

Step 2: For the current target date, the visibility criteria are checked for all TDRS satellites. All candidates are in the sensor's field of regard (FOR) but are not occulted by Earth's shadow. Hence, all candidates are theoretically observable. However, TDRS 5 does not meet the solar phase angle condition which, as its solar phase angle is $\approx 31.86°$ at the moment of the current target date. Hence, the step duration t_step is incremented for TDRS 5 and that object starts again from step 1 with an updated target date. TDRS 6 and 12, however, pass the visibility check at the current target date and proceed with step 3.

Step 3: By default, it was initially assumed that the allocation of the sensor towards the candidate's position shall take $t_{alloc,default} = 2\,\mathrm{s}$. This assumption is revisited in the slewing feasibility check and it shows that at the current target date the sensor does not manage to arrive in time at TDRS 6 nor at TDRS 12, since the actual allocation duration t_{alloc} takes approximately $105\,\mathrm{s}$ and $111\,\mathrm{s}$, respectively. Consequently, the remaining two satellites are also not suitable as tracking targets at the current target date, and their t_step needs to be incremented. At this point, all TDRS satellites need to revisit step 1 until a target date is found that meets all required conditions. The assigned target dates, at which the conditions are met for each candidate are:

- TDRS 5: - not trackable -
- TDRS 6: 22:03:06.620 UTC
- TDRS 12: 22:03:12.620 UTC

It shows that TDRS 5 was not trackable within the given time frame of $200\,\mathrm{s}$ as its solar phase angle remained below $33°$ and never exceeded the required minimum of $60°$.

Step 4: The simulated state update by means of an EKF reveals $d_{KL,TDRS6} = 31.09$ for TDRS 6 and $d_{KL,TDRS12} = 30.22$ for TDRS 12 as a measure for the information gained from a potential tracking task.

Step 5: The utility vectors assigned to the remaining candidates are:

$$\boldsymbol{u}_{track,TDRS6} = \begin{bmatrix} 31.09 \\ 106\,\mathrm{s} \end{bmatrix}$$

$$\boldsymbol{u}_{track,TDRS12} = \begin{bmatrix} 30.22 \\ 112\,\mathrm{s} \end{bmatrix}$$

Step 6: As can be seen from the previous step, the tracking utility vector of TDRS 6 dominates over the one of TDRS 12 in information gain and wasting time resources. As a result, TDR6 is selected to extend the MCTS by a new chance node. The state of the catalogue is propagated under the new tracking micro action, resulting in a new decision node, too.

Up until this point, the focus has been set on the selection of the micro action. In other words, the presented steps explain how to expand the decision tree. When applying the MCTS algorithm as detailed in Sec. 2.2, the pointing strategy presented in Tab. 4 is returned

Table 3. State and state covariance with reference to EME2000 on March 24, 2025, at 22:01:02.62 UTC

Satellite	State Vector (m, m/s)	State Covariance (m^2, m^2/s, m^2/s^2)
TDRS 5	$\begin{bmatrix} 30201985.91 \\ 28558626.98 \\ 7525934.90 \\ -2147.23 \\ 2137.16 \\ 511.17 \end{bmatrix}$	$\begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
TDRS 6	$\begin{bmatrix} -12013696.71 \\ 39168765.19 \\ 9914565.27 \\ -2947.53 \\ -853.43 \\ -207.52 \end{bmatrix}$	$\begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
TDRS 12	$\begin{bmatrix} -15563499.68 \\ 39106290.32 \\ 2639037.33 \\ -2856.77 \\ -1134.97 \\ -32.74 \end{bmatrix}$	$\begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 4. Schedule for tracking TDRS 5, 6 and 12 starting from March 24, 2025, at 22:03:06.620

Task ID	Target	Target Date	RA [deg]	Dec [deg]
0	TDRS 12	2025-03-24T22:01:02.620Z	111.6	3.1
1	TDRS 5	2025-03-24T22:01:13.620Z	111.1	3.2
2	TDRS 5	2025-03-24T22:01:44.620Z	111.3	3.1
3	TDRS 5	2025-03-24T22:02:15.620Z	111.4	3.1
4	TDRS 5	2025-03-24T22:02:46.620Z	111.5	3.1
5	TDRS 6	2025-03-24T22:03:48.620Z	106.2	14.9

4. CONCLUSION AND OUTLOOK

This work presents a proof-of-concept study demonstrating how Monte Carlo Tree Search (MCTS) can address multi-objective sensor tasking problems. The study introduced the multi-objective problem, emphasising that searching, Initial Orbit Determination (IOD), and tracking represent essential objectives in sensor management aimed at building and maintaining a full space object catalogue.

To tackle the complexity of determining optimal sensor pointing strategies, the approach was structured into two distinct optimisation aspects: macro actions (overall objectives) and micro actions (objective-specific strategies). As discussed in Sec. 2.3, macro-action optimisation employs the dominance principle to compare utility vectors. Future research will focus explicitly on applying MCTS to macro-action optimisation by defining and refining appropriate utility measures and dominance criteria.

Concerning micro-action optimisation within the searching objective, this study focused on the implementation of a stripe-scanning algorithm for systematically exploring a partial region around the Laplace ring to detect space objects. Currently, the stripe-scanning approach represents the only implemented method within the search micro-action space, resulting in a reward of one whenever this strategy is employed to follow a searching task. Future efforts will be dedicated to developing additional search algorithms and deriving specific micro-action rewards to

extend the MCTS framework.
With regard to the IOD objective, the Constrained Admissible Region (CAR) was successfully recovered and approximated using a Gaussian Mixture Model (GMM). Future work will aim to improve both the propagation of this distribution and the evaluation of measurement update value. A reward function for this micro-action space can be derived from information-theoretic measures such as the Kullback-Leibler divergence. The tracklet —resulting from a previous search task— that maximises information gain can then be selected as the most suitable candidate for the corresponding IOD task.
Finally, the micro-action space related to tracking was examined, too. The selection of a suitable tracking candidate follows an iterative six-step procedure. At the end of this process, each candidate is assigned a tracking utility vector. The candidate that provides the least dominated solution among all is selected as the tracking target, which extends the MCTS with a new chance and decision node. The utility vector is designed to reward high information gain and penalise long time gaps between consecutive measurements.

This proof-of-concept lays the groundwork for a decision-theoretic approach to autonomous sensor tasking. While the current implementation leaves several aspects unexplored, it provides a starting point for developing more refined and adaptive strategies in future work.

ACKNOWLEDGMENTS

The author acknowledges that ChatGPT (powered by OpenAI, GPT-4) was used for light editing of language. The technical content of this paper was generated by the human author.

REFERENCES

1. Kyle J DeMars and Moriba K Jah. Probabilistic initial orbit determination using gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 36 (5):1324–1335, 2013.
2. J-L Durrieu, J-Ph Thiran, and Finnian Kelly. Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4833–4836. Ieee, 2012.
3. Samuel Fedeler, Marcus Holzinger, and William Whitacre. Sensor tasking in the cislunar regime using monte carlo tree search. *Advances in Space Research*, 70(3):792–811, 2022.
4. Samuel J Fedeler. *Optical Sensor Tasking using Monte Carlo Tree Search*. PhD thesis, University of Colorado at Boulder, 2023.
5. Tim Flohrer. *Optical survey strategies and their application to space surveillance*. Schweizerische Geodätische Kommission, 2012.
6. Carolin Frueh, Hauke Fielder, and Johannes Herzog. Heuristic and optimized sensor tasking observation strategies with exemplification for geosynchronous objects. *Journal of Guidance, Control, and Dynamics*, 41(5):1036–1048, 2018.
7. Steven Gehly, Brandon Jones, and Penina Axelrad. Sensor allocation for tracking geosynchronous space objects. *Journal of Guidance, Control, and Dynamics*, 41(1):149–163, 2018.
8. RH Gooding. A new procedure for orbit determination based on three lines of sight: angles only. *NASA STI/Recon Technical Report N*, 94:21224, 1993.
9. Conor F Hayes, Mathieu Reymond, Diederik M Roijers, Enda Howley, and Patrick Mannion. Monte carlo tree search algorithms for risk-aware and multi-objective reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 37(2):26, 2023.
10. J Herzog, C Früh, and T Schildknecht. Build-up and maintenance of a catalogue of geo objects with zimsmart and zimsmart 2. 2010. In *61st International Astronautical Congress, Prague, Czech Republic. IAC-10 A*, volume 6.
11. Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
12. N Moretti, M Rutten, T Bessell, and B Morreale. Automated resident space object catalogue construction and maintenance using optical sensor management. In *Proc. International Astronautical Congress*, 2017.
13. Phuong Linh Ngo, Steve Gehly, Marco Langbroek, and Pieter Visser. Multi-objective sensor tasking of optical telescopes using monte carlo tree search. In *Proc. International Association for the Advancement of Space Safety*, 2024.
14. Laura Pirovano. *Cataloguing space debris: Methods for optical data association*. PhD thesis, University of Surrey, 2020.
15. Andres Rodriguez, Ronald Parr, and Daphne Koller. Reinforcement learning using approximate belief states. *Advances in Neural Information Processing Systems*, 12, 1999.
16. Manuel Schubert, Christopher Kebschull, Johannes Gelhaus, and Simona Silvestri. Evaluating sensor tasking strategies for object cataloging in geo. *Acta Astronautica*, 228:7–16, 2025.
17. Bob Schutz, Byron Tapley, and George H Born. *Statistical orbit determination*. Elsevier, 2004.
18. Jan Siminski. *Object correlation and orbit determination for geostationary satellites using optical measurements*. PhD thesis, Universitätsbibliothek der Universität der Bundeswehr München, 2016.
19. United States Strategic Command. Space-Track.org, 2025. URL `https://www.space-track.org/`. Accessed: 2025-03-23.
20. B-N Vo and W-K Ma. The gaussian mixture probability hypothesis density filter. *IEEE Transactions on signal processing*, 54(11):4091–4104, 2006.

21. Samuel Wishnek and Joshua Wysack. Leveraging fisher information to optimize observation scheduling for orbit determination. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, AMOS, Maui, Hawaii, USA*, 2023.

22. Yufeng Zhang, Jialu Pan, Li Ken Li, Wanwei Liu, Zhenbang Chen, Xinwang Liu, and Ji Wang. On the properties of kullback-leibler divergence between multivariate gaussian distributions. *Advances in Neural Information Processing Systems*, 36:58152–58165, 2023.