

Hypothesis-Driven Sensor Tasking for Space Domain Awareness

Ofer Dagan

University of Colorado, Boulder, USA

Tyler Becker

University of Colorado, Boulder, USA

Zachary N. Sunberg

University of Colorado, Boulder, USA

ABSTRACT

When human operators of cyber-physical systems, such as the Space Force’s Space Surveillance Tracking (SST) network, encounter surprising behavior, they often consider multiple hypotheses that might explain it. In some cases, taking information-gathering actions such as additional measurements or control inputs given to the system can help resolve uncertainty and identify the most accurate hypothesis. The task of optimizing these actions can be formulated as a belief-space Markov decision process that we call a *hypothesis-driven belief MDP*. Unfortunately, this problem suffers from the curse of history, similar to a partially observable Markov decision process (POMDP). To plan in continuous domains, an agent needs to reason over countless many possible action-observation histories, each resulting in a different estimate of the partially observable system state. The problem is exacerbated in the hypothesis-driven context, since each action-observation pair spawns several estimates resulting from the different hypotheses. This paper considers the case in which each hypothesis corresponds to a different dynamic model in an underlying POMDP. We present a new belief MDP formulation that: (i) enables reasoning over multiple hypotheses, (ii) balances the goals of determining the (most likely) correct hypothesis and performing well in the underlying task, and (iii) can be solved with sparse tree search.

1. INTRODUCTION

Over the past few years there has been an exponential increase in the number of resident space objects (SOs), from about 19,000 to 30,000 objects in 2017 to 2024 [15], with an expected increase to hundreds of thousands in the next few years [8]. This poses a significant challenge from a space domain awareness (SDA) perspective. In addition to the task of maintaining a catalog of orbits for each of the SOs, an increasing number of anomalous situations must be investigated.

Consider a scenario in which, during routine sensor tasking operation for catalog maintenance according to a fixed, already optimized, schedule, an anomaly in the orbit of one of the SOs is detected (Figure 1). As a result, the operator – potentially without access to the planning algorithm or model of the sensor tasking system – is interested in exploring several hypotheses that might explain it: the first is that the OOI deployed solar panels, which will result in an increase in drag and might change its orbit; the second is an engine misfire, which instantaneously changes the velocity of the OOI; and the third is a change in the mass of the OOI, potentially from a change in structure. For each of these hypotheses, the operator formulates a new dynamics function, which captures the possible anomaly explanation.

Traditional formulations of the problem aim at capturing several different objectives, such as catalog maintenance [4], maneuver detection [10], or estimating control modes [3]. However, these objectives are not always aligned, as some of them require repeated observations of specific OOI, which is especially challenging in a resource-constrained environment. Moreover, these often need to be defined a priori and cannot be added to an existing plan in an ad hoc manner. In recent years, there has been a shift toward hypothesis-driven approaches, which quantify questions, or hypotheses, as testable quantities, on which information can be



Fig. 1: A team of human operators monitoring SOs, when one SO suddenly changes orbit. The operators want to collect data that might help explain the reason for the change.

gathered. In [9], a formal framework for reasoning using Dempster-Shafer theory is proposed, where it is suggested to incorporate ambiguity through evidential reasoning. Our approach focuses on a Bayesian formulation of the problem, in the form of multi-hypothesis planning, where different hypotheses are converted to a probability distribution, allowing the decision-maker to make an informed decision based on the gathered data.

To resolve the uncertainty over the correct hypothesis, we augment the original sensor tasking schedule, the “base problem”, and form a new planning problem – the “multi-dynamics hypothesis (MDH) planning” problem, with objectives that compete with the objectives of the original tasking problem. For example, the objective of the catalog maintenance schedule is to minimize the uncertainty over the entire catalog, while in the MDH case, one of the objectives is to reduce uncertainty over a set of hypotheses over one specific SO – the object of interest (OOI). This might lead to tasking sensors to gather information over the OOI instead of scanning the entire catalog. This paper explores the hypothesis-driven sensor tasking problem. We formulate the problem as a belief-space Markov decision process that we name a hypothesis-driven belief MDP. This enables reasoning over multiple hypotheses while allowing tractable solutions using existing sparse tree search algorithms such as Monte-Carlo tree search (MCTS). We focus on hypotheses that stem from different dynamic models that might cause an anomaly in the OOI orbit and explore different reward functions to balance the goals of determining the most likely hypothesis while performing well with respect to the original tasking schedule.

2. BACKGROUND

2.1 Belief-MDP

The Markov Decision Process (MDP) is a fundamental framework in sequential decision-making under uncertainty. A particular MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where \mathcal{S}, \mathcal{A} are the sets of all possible states and actions, respectively. $\mathcal{T}(s, a, s')$ is a stochastic state transition model, which defines the probability of transitioning to state $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ after taking action $a \in \mathcal{A}$. The reward function $\mathcal{R}(s, a)$ determines the immediate reward for taking action a at state s , and $\gamma \in [0, 1)$ is a discount factor. A solution for an MDP is an optimal policy $\pi^*(s)$ that maximizes the discounted sum of rewards,

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right]. \quad (1)$$

An extension to MDP is the partially observable MDP (POMDP), where the true state of the system is unknown. In these problems the agent collects noisy measurements o to estimate the state, maintaining a *belief* $b(s)$ over states $s \in \mathcal{S}$. The belief summarizes the history h_t of all actions taken and observations received up to and including time step t and starting from a prior belief b_0 ,

$$b(s) = p(s_t = s | h_t), \quad \text{where } h_t = (b_0, a_0, o_1, a_1, \dots, o_t). \quad (2)$$

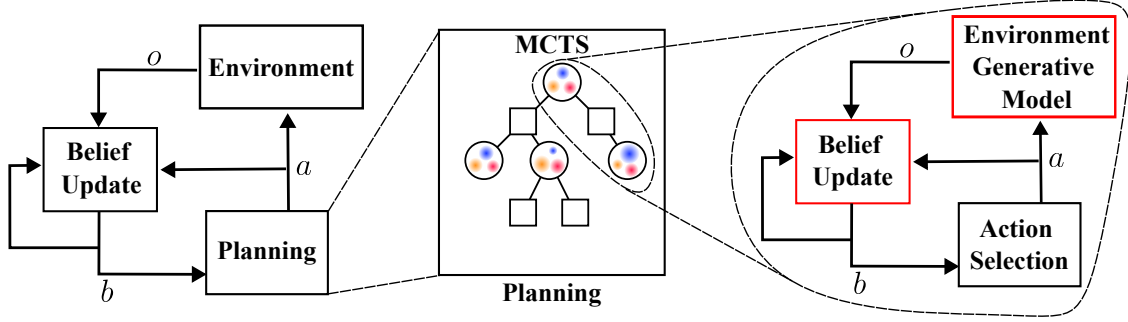


Fig. 2: Planning - Acting - Observing - Estimating loop diagram and how it translates to the MCTS algorithm.

Formally, a POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, Z, \mathcal{R}, \gamma)$, where \mathcal{O} is the set of all possible observations, and $Z(a, s', o)$ is the observation function, defined as the conditional probability of seeing observation $o \in \mathcal{O}$ after taking action a and reaching state s' .

A fundamental issue of POMDPs is that the reward function depends directly on the state of the system. While optimal POMDP policies may reduce uncertainty if such a reduction helps to maximize the reward, there are many cases, such as hypothesis investigation in SDA, where there is an external reason for reducing uncertainty that cannot easily be expressed using a state-based reward. In such cases, a reward that can explicitly penalize uncertainty is needed [1]. For example, in sensor tasking problems, we want a plan that prioritizes observations to support decisions and objectives that cannot be expressed with a reward function on the physical states of the SOs. Such observations should be chosen to maximize some measure of information, e.g. entropy reduction.

A solution to this problem is to reason about belief-states instead of states. A belief-MDP is a transformation of a POMDP to belief space, defined by the tuple $(\mathcal{B}, \mathcal{A}, \tau, \rho, \gamma)$. Where \mathcal{A} and γ are the same as before, \mathcal{B} is the set of possible belief states $b \in \mathcal{B}$, $\tau(b, a, b')$ is the belief transition model [12] from the prior belief b to the posterior belief b' , which accounts for observations using Bayes' rule,

$$\tau(b, a, b') = p(s'|s, a) = \int_{o \in \mathcal{O}} p(s'|a, s, o) \cdot p(o|a, s) do, \quad (3)$$

and $\rho(b, a)$ is the belief-dependent immediate reward [1]. Where $\rho(b, a)$ can directly relate some measure of uncertainty, such as Shannon's entropy or Kullback-Leibler divergence, or indirectly via the expected reward, when the state s is distributed according to b ,

$$\mathbb{E}_{s \sim b} [\mathcal{R}(s, a)] = \int_{s \in \mathcal{S}} b(s) \mathcal{R}(s, a) ds. \quad (4)$$

2.2 Monte Carlo Tree Search

There are several terminologies used to describe the phases of sequential decision processes in the literature. One common set of terms for the steps needed to make a decision is the OODA loop: observe – orient – decide – act. In this paper, we choose terms that relate to Bayesian estimation and decision making, namely plan (decide) – act – observe – estimate (orient). The left diagram in Fig. 2 describes a cycle in an online planning loop, where an agent plans based on its prior belief b , chooses an action a , interacts with the environment, which omits a measurement o . Based on the action-measurement pair (a, o) , and given its uncertain models of the environment, the agent updates its belief,

$$b'(s') = p(s'|h_t) \propto p(o|a, s') \cdot \int_{s \in \mathcal{S}} b(s) \cdot p(s'|s, a) ds. \quad (5)$$

Monte Carlo tree search (MCTS) is a widely used technique to solve various reasoning problems, and in recent years has been integrated with reinforcement learning systems such as AlphaZero [18] and AlphaStar

[20]. In the context of Belief-MDP problems, at each planning step, MCTS builds a tree, consisting of belief (circle) and action (square) nodes, as shown in Fig. 2 (middle), up to a predefined horizon. The result of the simulation is the (current step) action that maximizes the belief-action value function (Q-function), when following policy π ,

$$Q(b, a) = \mathbb{E}_{s \sim b} \left[\sum_{t=0}^{\infty} \gamma^t \rho(b_t, a_t) \middle| b_0 = b, a_0 = a, a_t = \pi(b_t) \right]. \quad (6)$$

The Q-function represents the expected return when starting from a belief b , taking an action a and then following a policy π . MCTS aims to estimate the optimal Q-function, Q^* , from which the optimal policy can be recovered with $\operatorname{argmax}_a Q^*(b, a)$.

Since the MCTS algorithm is not the focus of this paper, we now discuss the elements of the algorithm that are relevant in the context of this paper and refer the interested reader to the survey by [2] for more details about the MCTS algorithm and to the documentation of the POMDPs.jl framework [5] for implementation details.

Fig. 2(right) shows a process diagram of one expansion of the tree – from a prior belief b to a posterior belief b' . Note that after taking an action a from belief b , there are many possible posterior beliefs b' , each conditioned on a different possible observation o . The two MCTS elements pertinent to this paper are the environment model and the Bayesian belief update. In the context of POMDPs, an environment generative model $G(\mathcal{P}, s, a)$ is a function that generates the next state s' , an observation o , and a reward r , given a problem tuple, \mathcal{P} , a prior state s , as described in Algorithm 1.

Assuming a Bayesian approach, a belief update refers to the process in which a prior distribution is propagated according to the system dynamics and then updated according to a measurement, as described in eq. (5). While the key update steps are general, the implementation differs based on the type of underlying distribution, its representation, and the functions \mathcal{T} and Z . Further discussion regarding the type of updater (estimator) suited for the multi-dynamics hypothesis sensor tasking problem is given in Section 4.3.

Algorithm 1 Environment Generative Model $G(\mathcal{P}, s, a)$

- 1: **Input:** $s, a, \mathcal{P} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, Z, \mathcal{R}, \gamma\}$
 - 2: $\mathcal{T} = \mathcal{P}.\mathcal{T}, Z = \mathcal{P}.Z, \mathcal{R} = \mathcal{P}.\mathcal{R}$
 - 3: $s' \sim \mathcal{T}(s, a, s')$
 - 4: $o \sim Z(a, s', o)$
 - 5: $r \leftarrow \mathcal{R}(s, a, s', o)$
 - 6: **return** (s', o, r)
-

2.3 Integer Linear Programming for Catalog Maintenance Tasking

As our base problem, we consider catalog maintenance, one of the core problems in space domain awareness. It refers to the problem of tasking a set of sensors to maintain custody over an existing “catalog” of space objects. This multi-sensor tasking problem can be formulated as a POMDP and has been solved via MCTS [6], deep reinforcement learning [17], or policy-gradient model predictive control [19]. In this paper, we choose to generate the optimal base plan π^* , a schedule for tasking observers to SO at different time slots, using integer linear programming (ILP).

The base ILP [14] approach aims to allocate sensors to space objects over a fixed period, maximizing the minimum observation count across all satellites. This ensures a fair distribution of observations, minimizing the disparity in how often each satellite is observed. In doing so, the approach satisfies the catalog

maintenance objective for SDA. Formally, the ILP is given by

$$\begin{aligned}
& \text{maximize} && \tau \\
& \text{subject to} && X_{ijt} \in \{0, 1\}^{I \times J \times T} \\
& && X_{ijt} \leq O_{ijt} \quad \forall i, j, t \\
& && \tau \leq \sum_{j,t} X_{ijt} \quad \forall i \\
& && \sum_i X_{ijt} \leq 1 \quad \forall j, t.
\end{aligned} \tag{7}$$

Here X_{ijt} is a binary 3-dimensional control variable representing whether observer j observes object i at time step t , and O_{ijt} represents whether or not observer j is able to observe object i at time t . Similar ILP formulations have been widely applied in scheduling and sensor assignment problems [21, 13].

3. PROBLEM STATEMENT

Consider again the multi-hypothesis sensor tasking scenario described in the introduction, where a human operator, monitoring an SDA system, operating according to a fixed, already optimized plan π^* , encounters a surprising behavior of one of the objects, the OOI. In many cases, the operator doesn't have access to the OOI's underlying transition model \mathcal{T} , so to try and explain what they observe, they form a set of $n_{\mathcal{H}}$ hypotheses, or alternatives to the underlying (nominal) model. We define the set of questions of whether model i is correct or not as the set of hypotheses \mathcal{H} , where each hypothesis \mathcal{H}_i corresponds to transition model $\mathcal{T}_i(\theta_i)$, parameterized by the random variable (RV) θ_i . For example, they hypothesize that the OOI deployed solar panels, which would affect its trajectory due to drag, parameterized by $\theta_1 \sim p(\theta_1 | \mathcal{H}_1)$, so the new transition model is $\mathcal{T}_1(\theta_1)$. The agent's task is to repeatedly decide whether to keep the original sensor assignment (according to π^*), or to change it in order to infer the most likely hypothesis.

Formally, we define the augmented problem $\bar{\mathcal{P}}$, of deciding which hypothesis is correct, while performing well in the underlying problem \mathcal{P} (with respect to π^*), as the multiple-dynamics hypothesis belief MDP (MDH-BMDP), which allows us to explicitly reward uncertainty reduction over the hypotheses' beliefs,

$$\bar{\mathcal{P}} = (\bar{\mathcal{B}}, \bar{\mathcal{A}}, \bar{\tau}(\bar{\theta}), \bar{\rho}, \gamma). \tag{8}$$

Here $\bar{\mathcal{B}}, \bar{\mathcal{A}}, \bar{\tau}(\bar{\theta}), \bar{\rho}$, and γ are as defined in Section 2.1; and the bar indicates augmented quantities that will be detailed in the next section.

Given the planning problem $\bar{\mathcal{P}}$, we seek an optimal policy $\bar{\pi}^*$, that maximizes the discounted sum of rewards,

$$\bar{\pi}^* = \operatorname{argmax}_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \bar{\rho}(\bar{b}_t, \bar{a}_t) \right]. \tag{9}$$

Where the optimal policy must balance between two potentially competing requirements – deciding which hypothesis \mathcal{H}_i is (most likely) correct, while still performing well with respect to the optimal policy π^* of the original underlying problem \mathcal{P} . Note that in cases where the two requirements align, acting according to $\bar{\pi}^*$ should result in the same performance in the underlying problem as if the agent were acting based on the original policy π^* .

4. HYPOTHESIS-DRIVEN SENSOR TASKING

Consider an underlying multi-sensor tasking problem \mathcal{P} , with n_{obs} observers (sensors) and n_{so} space objects, which can be posed as a POMDP [6]. We define \mathcal{P} by the tuple,

$$\mathcal{P} = (\mathcal{S}_x, \mathcal{A}_x, \mathcal{T}, \mathcal{O}, Z, \mathcal{R}_x, \gamma), \tag{10}$$

, where \mathcal{S}_x is the state space of the underlying system, \mathcal{A}_x is the underlying action space, \mathcal{T} , \mathcal{O} , Z are the transition function, the observation space, and the observation function, respectively. An optimal solution for \mathcal{P} is a policy $\pi^*(s_x)$, which maximizes the cumulative sum of discounted reward $\mathcal{R}_x(s_x, a_x)$ according to eq. (1), with a discount factor γ .

Typically, POMDP problems reason about one transition model \mathcal{T} , and one observation function Z associated with the problem definition tuple \mathcal{P} . This section describes the solution approach for the MDH-planning problem that enables the use of existing algorithms (e.g., MCTS) to solve the hypothesis-driven problem, balancing between the original plan objective, and determining which hypothesis is most likely correct.

There are two key design choices in our approach to the problem. First, we model the augmented problem $\bar{\mathcal{P}}$ as a belief-MDP, which enables rewarding uncertainty reduction over the hypothesis belief. Second, we frame each hypothesis as a POMDP \mathcal{P}_i , which allows flexibility in the definition of hypotheses.

4.1 The MDH-BMDP

Given the underlying POMDP problem \mathcal{P} (equation (10)), and a set of $n_{\mathcal{H}}$ hypotheses that we want to consider, which differ by their transition model,

$$\{\mathcal{H}_1(\mathcal{T}_1), \mathcal{H}_2(\mathcal{T}_2), \dots, \mathcal{H}_{n_{\mathcal{H}}}(\mathcal{T}_{n_{\mathcal{H}}})\}, \quad (11)$$

where each transition model i is parameterized by θ_i . We define the elements of the MDH-BMDP tuple $\bar{\mathcal{P}} = (\bar{\mathcal{B}}, \bar{\mathcal{A}}, \bar{\tau}(\bar{\theta}), \bar{\rho}, \gamma)$ in equation (8) as follows:

- $\bar{\mathcal{B}} = \mathcal{B}_x \times \mathcal{B}_\theta \times \mathcal{B}_{\mathcal{H}}$ is the joint belief space, based on the underlying state space \mathcal{S}_x , the parameter space θ , and the hypotheses state space $\mathcal{S}_{\mathcal{H}} = \mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_{n_{\mathcal{H}}}$. As a result a belief state $\bar{b} \in \bar{\mathcal{B}}$ is given by,

$$\bar{b}(\bar{s}) = p(s_x, \theta, s_{\mathcal{H}}) = \underbrace{p(s_{\mathcal{H}})}_{b_{\mathcal{H}}} \cdot \underbrace{p(s_x, \theta | s_{\mathcal{H}})}_{b_{x|\mathcal{H}}}. \quad (12)$$

Where $b_{\mathcal{H}}$ is a categorical distribution, and with slight abuse of notation, $b_{x|\mathcal{H}}$ is the conditional distribution of the underlying state and θ given the hypothesis \mathcal{H} . In our sensor tasking problem, we model it as a Gaussian distribution. Thus, the joint belief \bar{b} is a mixture of distributions, weighted by the probability of each hypothesis \mathcal{H}_i being correct. To allow the agent to decide when to commit to a certain hypothesis, i.e., declaring which hypothesis it believes is most likely correct, we add a deterministic state that we name *resolved*. The resolved state can take values of $0 : n_{\mathcal{H}}$, where 0 indicates no commitment, and $1 : n_{\mathcal{H}}$ indicates which hypothesis it decides to commit to.

- $\bar{\mathcal{A}}$ is the action space, where $\bar{a} \in \bar{\mathcal{A}}$ is the tuple $(a_x, a_{\mathcal{H}})$. Here $a_x = 0 : n_{obs}$, with 0 meaning leaving the plan as is, and otherwise changing the plan of observer $j = 1 : n_{obs}$ to take an observation of the OOI. $a_{\mathcal{H}}$ gives the agent the option to decide whether to commit to a decision regarding which hypothesis is correct ($a_{\mathcal{H}} = 1$), or not ($a_{\mathcal{H}} = 0$).
- $\bar{\tau}(\bar{b}, \bar{a}, \bar{b}')$ is the transition function, defining how to update a new joint belief \bar{b}' , from \bar{b} , after taking an action \bar{a} . More details about $\bar{\tau}$ are given in Section 4.3, but notice that $\bar{\tau}$ depends on the underlying transition model \mathcal{T} , observation space \mathcal{O} , and observation function Z , and the set of hypotheses transition models $\mathcal{T}_1, \dots, \mathcal{T}_{n_{\mathcal{H}}}$.
- $\bar{\rho}$ is the joint belief-dependent reward function defined later in Equation (15). It transitions the underlying state-dependent reward R_x to a belief-dependent reward using Equation (4) and adds the hypothesis reward $\rho_{\mathcal{H}}$. Further discussion on the reward function is given in Section 4.4.
- γ is the discount factor, inherited from the underlying POMDP problem \mathcal{P} .

4.2 Constructing the MDH-BMDP

The key idea in the implementation of the MDH-BMDP framework is to define an array \bar{P} of $n_{\mathcal{H}}$ ‘‘hypothesis conditioned’’-POMDPs \mathcal{P}_i ,

$$(\mathcal{S}_x, \mathcal{A}_x, \mathcal{T}_i, \mathcal{O}, Z, \mathcal{R}_x, \gamma), \quad (13)$$

where, \mathcal{T}_i is based on the underlying POMDP problem \mathcal{P} , but with a dynamics function f_i corresponding to hypothesis \mathcal{H}_i . Note the subtle difference between the dynamics function f_i and the transition model \mathcal{T} – while f_i describes the equations of motion, \mathcal{T} depends on f_i and represents a distribution over possible states s'_x . The rest of the tuple elements are inherited from \mathcal{P} . To be able to use the different POMDPs \mathcal{P}_i with existing solvers within the POMDPs.jl ecosystem [5], construction of the MDH-BMDP is performed in three key steps:

1. Constructing a single hypothesis POMDP \mathcal{P}_i for each dynamic model f_i . For example, f_1 is the nominal equations of motion for satellite dynamics, with minimal drag, i.e., θ close to 0; and f_2 has higher drag, i.e., assuming some distribution of θ that is larger than the nominal.
2. Transforming each POMDP \mathcal{P}_i into a belief-MDP \mathbf{P}_i by changing the states to belief-states, adding the parameter θ to the belief-state, and adding an updater. For example, in our SDA problem, we choose to model our belief over the 3D position and velocity SO state, s_x , and the parameter θ , as a Gaussian, $b_{x|\mathcal{H}} \sim \mathcal{N}([x, \theta]; \mu, \Sigma)$. For our updater we choose the unscented Kalman filter (UKF) [11].
3. Constructing the MDH-BMDP $\bar{\mathcal{P}}$,

$$(\bar{\mathbf{P}}, \bar{\mathcal{B}}, \bar{\mathcal{A}}, \bar{\tau}(\bar{\theta}), \bar{\rho}, \gamma), \quad (14)$$

where we added a tuple of BMDPs $\bar{\mathbf{P}} = (\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n_{\mathcal{H}}})$ to the definition in Equation (8).

This structure of the MDH-BMDP allows for the belief update of the multi-hypothesis belief \bar{b} , as discussed in the next section.

4.3 Belief Transition Model

Planning in belief-MDPs requires reasoning over belief-states instead of over states, thus a generative model as defined in Algorithm 1 has to be modified to propagate the belief b and not the state s , as described in Algorithm 2. Propagating a prior belief b to a posterior belief b' in a Bayesian belief updater (estimator) includes two main steps: prediction and correction (Equation 5). In the prediction step, the belief is propagated according to the transition model $\mathcal{T}(s, a, s') = p(s'|s, a)$. In the correction step, also known as measurement update, Bayes' rule is used to fuse an observation according to the model's observation model $Z(a, s', o) = p(o|s', a)$. However, since we are reasoning over a belief state, we don't have a unique state s' to sample an observation, as in line 4, Algorithm 1. Instead, we need to sample a state from the belief, by first sampling a hypothesis state $s_{\mathcal{H}} = \mathcal{H}_i$ out of $b_{\mathcal{H}}$, and then an underlying state s_x and the parameter θ are sampled from the conditional distribution $p(s_x|s_{\mathcal{H}})$ (lines 4-5). An observation is generated using Algorithm 1, called on the POMDP $\mathcal{P}_{i=s_{\mathcal{H}}}$, corresponding to the sampled hypothesis state $s_{\mathcal{H}}$.

We can see here that the structure of the MDH-BMDP allows for the use of the hypothesis-relevant transition and observation models within the environment generative model G . Similar use is done within the update and reward calls on lines 7-8, respectively.

In the case of MDH-BMDP, the joint belief \bar{b} is defined as in Equation (12), where $b_{\mathcal{H}}$ is a categorical distribution, Thus \bar{b} is a mixture of distributions. The belief updater is then of multiple-model (MM) type, with the key steps described in Algorithm 3. More details regarding different versions of MM estimators can be found in [16], where for the sensor tasking problem, where we model the belief over the underlying state as a Gaussian, we implement a Gaussian mixture (GM) estimator.

Lastly, the deterministic “resolved” state needs to be updated (lines 10 – 13). This update is determined by the agent's action and the prior state – if the state is not yet resolved, and the agent decides to commit ($a_{\mathcal{H}} = 1$), the state is updated to the maximum likelihood estimate of the hypothesis (with an appropriate tie-breaking method).

4.4 Reward Function

The last element in the definition of the MDH-BMDP $\bar{\mathcal{P}}$ is the belief-reward function $\bar{\rho}$. The goal is to construct a reward function that will balance two requirements: (i) making a decision regarding the most probable hypothesis, and (ii) solving the original underlying problem \mathcal{P} with minimal effect on the original policy.

Algorithm 2 Belief Transition Model

```
1: Input:  $\bar{b}, \bar{a}, \bar{\mathcal{P}} = (\bar{\mathbf{P}}, \bar{\mathcal{B}}, \bar{\mathcal{A}}, \bar{\tau}(\bar{\theta}), \bar{\rho}, \gamma)$ 
2:  $\bar{\rho} = \bar{\mathcal{P}} \cdot \bar{\rho}$ 
3:  $b_x = \bar{b}.b_x, b_{\mathcal{H}} = \bar{b}.b_{\mathcal{H}}$ 
4:  $s_{\mathcal{H}} \sim b_{\mathcal{H}}$ 
5:  $s_x \sim b_x[s_{\mathcal{H}}]$ 
6:  $o \leftarrow G(\mathbf{P}_{s_{\mathcal{H}}}, s_x, a_x)$  ▷ Algorithm 1
7:  $\bar{b}' \leftarrow \text{update}(\bar{\mathcal{P}}, \bar{b}, \bar{a}, o)$  ▷ Algorithm 3
8:  $r \leftarrow \bar{\rho}(\bar{b}, \bar{a}, \bar{b}', o)$ 
9: return  $(\bar{b}', r)$ 
```

Algorithm 3 MM Belief Update

```
1: Input:  $\bar{\mathcal{P}}, \bar{b}, \bar{a}, o$ 
2:  $b_x = \bar{b}.b_x, b_{\mathcal{H}} = \bar{b}.b_{\mathcal{H}}$ 
3: for  $i = 1, \dots, n_{\mathcal{H}}$  do
  // Update conditional belief over the underlying state:
4:    $\text{updater} = \bar{\mathcal{P}} \cdot \mathbf{P}[i]$ 
5:    $b'_x[i] \leftarrow \text{predict}(\text{updater}, b_x[i], a_x)$ 
6:    $\bar{b}'.b'_x[i] \leftarrow \text{correct}(\text{updater}, b_x^-[i], a_x, o)$ 
  // Correct belief over hypotheses:
7:    $b'_{\mathcal{H}}[i] \leftarrow b_{\mathcal{H}}[i] \cdot \int_{x \sim b'_x[i]} p(o|\mathcal{H}_i, x) dx$ 
8: end for
9:  $\bar{b}'.b'_{\mathcal{H}} \leftarrow b'_{\mathcal{H}} / \sum_i b'_{\mathcal{H}}[i]$  ▷ renormalize
10:  $\bar{b}'.resolved = \bar{b}.resolved$ 
11: if  $a_{\mathcal{H}} == 1$  &  $!\bar{b}.resolved$  then
12:    $\bar{b}'.resolved = \max(\bar{b}'.b'_{\mathcal{H}})$ 
13: end if
14: return  $\bar{b}'$ 
```

Similar to [7], we define the reward function as the convex sum of the expected state-action reward and a belief-dependent reward,

$$\bar{\rho}(\bar{b}, \bar{a}) = (1 - w) \cdot \rho_x(b_x, a_x) + w \cdot \rho_{\mathcal{H}}(b_{\mathcal{H}}, a_{\mathcal{H}}), \quad (15)$$

with $w = [0, 1]$ is the weight parameter to balance between the two requirements, and

$$\rho_x(b_x, a_x) = \sum_{i=1:n_{\mathcal{H}}} b_{\mathcal{H}}[i] \cdot \rho_x(b_{x|\mathcal{H}[i]}, a_x) = \sum_{i=1:n_{\mathcal{H}}} b_{\mathcal{H}}[i] \cdot \int b_{x|\mathcal{H}[i]}(s_x) \mathcal{R}_x(s_x, a_x) ds_x. \quad (16)$$

That is, the underlying task reward is the weighted sum of the rewards of the hypothesis-conditioned belief-state.

We now focus our attention on the hypothesis-related reward function $\rho_{\mathcal{H}}(b_{\mathcal{H}}, a_{\mathcal{H}})$. The agent is expected to generate a new plan, or to make changes to the original plan, that balances between the underlying system objectives, and maximizes the probability of determining which hypothesis is correct. In a realistic scenario, the human user or operator might want to make a decision within a time limit T .

To minimize uncertainty, or maximize probability, a natural choice for the belief-dependent reward is to use some measure of information [1], [7], e.g., negative Shannon's entropy,

$$\rho_{\mathcal{H}}(b_{\mathcal{H}}, a_{\mathcal{H}}) = \sum_{s_{\mathcal{H}} \in \mathcal{S}_{\mathcal{H}}} b_{\mathcal{H}}(s_{\mathcal{H}}) \log b_{\mathcal{H}}(s_{\mathcal{H}}). \quad (17)$$

However, while this type of reward function can incentivize information-gathering actions that reduce entropy, thus trying to “push” one of the hypotheses' probabilities in the direction of 1, it does not explicitly set a time limit on making a decision. We suggest a new simple sparse reward function, which we name “commit-reward”, that explicitly accounts for making a decision of which hypothesis is correct within time T . We add

a commit action to the action space, allowing the agent to decide when to commit to a certain hypothesis, thus autonomously balancing between the hypothesis and the underlying system’s rewards.

$$\rho_{\mathcal{H}}(b_{\mathcal{H}}, a_{\mathcal{H}}) = \begin{cases} \max(b_{\mathcal{H}}) & \text{if } a_{\mathcal{H}} = 1 \ \& \ t \leq T \ \& \ !b.\text{resolved} \\ 0.0 & \text{otherwise} \end{cases} \quad (18)$$

In other words, the agent can collect the hypothesis reward if it commits to an hypothesis by the time limit T . It can only collect this reward once, if it hasn’t committed yet ($b.\text{resolved} == 0$). The hypothesis reward then equals the maximum probability, as there is no reason to commit to any other hypothesis than the one that is most likely. Lastly, note that the commit-reward is a finite-horizon reward (commit by time T), while the underlying reward ρ_x is, in general, an infinite-horizon reward. To compute the Q-value function (Equation (6)), we multiply the commit reward by γ^{-t} , to cancel the effect of discounting on that part of the joint reward.

5. EXPERIMENTS

As our base problem we consider catalog maintenance, one of the core problems in space domain awareness. Consider again a scenario where, during routine operation of the sensor tasking schedule, the “base” optimal policy, or plan, π^* , an anomaly in the orbit of one of the SOs is detected. As a result, the operator is interested in determining the origin of the anomaly, e.g., due to an engine misfiring, a deployment of solar panels, or possibly an intentional maneuver, which defines our augmented multi-hypothesis problem \bar{P} .

To demonstrate how MDH-BMDP is used for multi-hypothesis sensor tasking, we consider the following scenario: there are 5 SOs in low Earth orbit (LEO), and an anomaly is detected on one of them, the OOI, which spawns several hypotheses regarding the dynamic model: the first is the nominal model, which states that nothing happened and the OOI continues its orbit according to the original dynamics model. Other models, as will be described later, consider dynamic models (hypotheses) that include drag, engine misfire (Δv), and a change in mass. Given a nominal catalog maintenance sensor-tasking plan \mathcal{P} , generated with ILP (Section 2.3), for 3600 seconds with a time step of 60 seconds, the MDH-BMDP problem $\bar{\mathcal{P}}$ seeks to determine the correct hypothesis within 2400 seconds, with minimal change to the original plan. In this problem, when transitioned into an MDH-BMDP, the belief over the OOI state is approximated as a Gaussian distribution and propagated using an unscented Kalman filter (UKF) [11], one for each hypothesis.

In hypothesis-driven planning, the planning agent optimizes for actions that both reduce uncertainty and still perform well in the underlying problem. The simulation analysis then focuses on parameters relating to decision quality – was the decision correct, and was it made in time, and how much the performance of the underlying problem was affected – as measured by the state-action dependent cumulative discounted reward (“base reward”).

5.1 Commit vs. Entropy Reward

The multi-hypothesis sensor tasking problem is a type of information gathering problem, where we aim to motivate the agent to deviate from the original plan and take observation actions to the OOI. However, we have competing objectives – on the one hand, we wish to increase the probability of the correct hypothesis by the “deadline” of 2400 sec, on the other hand, we wish to minimize the changes to the original catalog maintenance schedule. To account for changes in the original catalog maintenance plan, each additional OOI observation action incurs a penalty of -1 , i.e., $\mathcal{R}(s_x, a_x) = -1$ if $a_x! = 0$.

This section focuses on: (i) exploring two possible hypothesis reward functions $\rho_{\mathcal{H}}$: the Shannon entropy function (eq. (17)), which is commonly used in information gathering problems, vs. our new “commit” reward function (eq. (18)); and (ii) choosing the reward weight w , which balances the competing objectives. In the simulation scenario, there are 3 hypotheses: in the first, the correct hypothesis, the OOI has minimal drag, in the second and third, it is hypothesized that the OOI deployed medium and large solar panels, respectively. The parameter θ of the two new transition models $\mathcal{T}(\theta)$ represents the solar panels’ area as a normally-distributed continuous random variable (RV).

Figure 3 presents averaged results from 100 Monte-Carlo (MC) simulations. For each reward, we show the Pareto front as a function of the weight w (Figure 3(a)-(b)) of the two objectives as quantified by the base

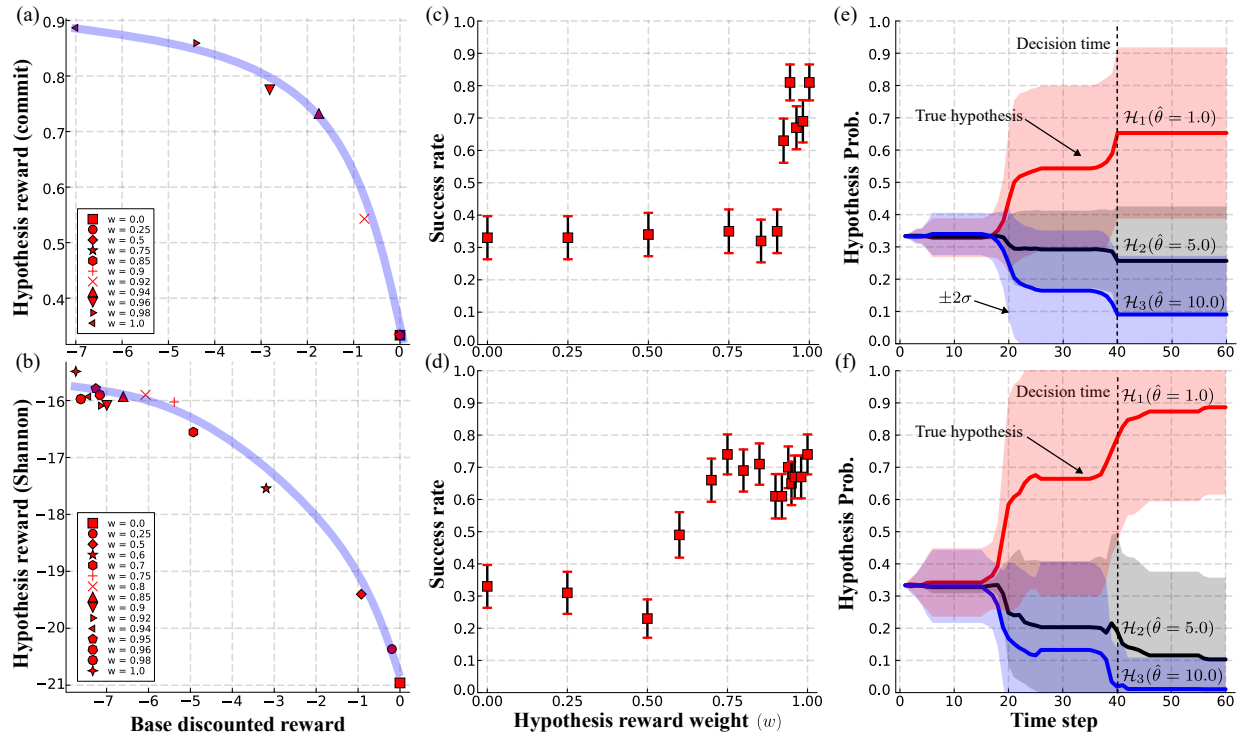


Fig. 3: Pareto front, success rate, and hypothesis probabilities trajectory for the commit (upper row) and entropy (lower row) reward functions.

discounted reward and the hypothesis reward. From Figure 3(a), we can see the trade-off between the two objectives: as w decreases from the pure hypothesis objective ($w = 1$), the base reward increases, due to fewer changes to the original plan. At a certain point, for $w < 0.92$, the penalty for changing the plan is too large, so the agent chooses to keep the original plan, and does not take observations to the OOI. The belief over the hypotheses does not change from the prior, equally probable belief, and the base reward is 0. The picture is similar for Shannon's entropy reward in Figure 3(d). However, due to the different ratio between the rewards, the effect of w is different. Here we see that values larger than 0.85 give very similar results, and might be considered statistically the same. For $w \leq 0.85$ we see that the agent starts to make fewer changes to the plan.

Since all points on the Pareto front are Pareto-optimal, improving one objective comes at a cost for the other objective. The human operator then needs to decide what is a reasonable trade-off from their perspective. In this example, we suggest using a measure of success for determining which hypothesis is correct, in time, i.e., by the 2400 sec (40 time-steps) deadline. Figure 3(c)-(d) present the success rate across 100 simulation, where success is measured by the maximum likelihood estimate of the correct dynamic model, as measured at the deadline. For example, for the commit reward function, $w = 0.94$ has a high success rate (Figure 3(c)), while a relatively high base reward (Figure 3(a)). Notice that for $w = 0.92$, a small gain in the base reward means about 15% decrease in the success rate. From similar reasons, for the entropy reward, we would choose $w = 0.75$ as the weight value.

Lastly, there are two key differences when comparing the two reward functions: first, the base discounted reward, which reflects how much we deviated from the original plan. With the commit reward, for $w = 0.94$ the base reward is higher than -2 , reflecting 6.8 changes to the original plan on average. While with the entropy reward, for $w = 0.75$, the base reward is about -5.5 , reflecting 26 changes to the original plan. The second point is that when comparing the two reward functions can be seen from Figure 3(e)-(f), showing the probability of each hypothesis as a function of time steps for $w = 0.94$ and $w = 0.75$, respectively. We can see that for the commit reward function, the probabilities do not change after the decision time limit of time step 40, since the agent already committed to one hypothesis, and is not motivated to change the plan

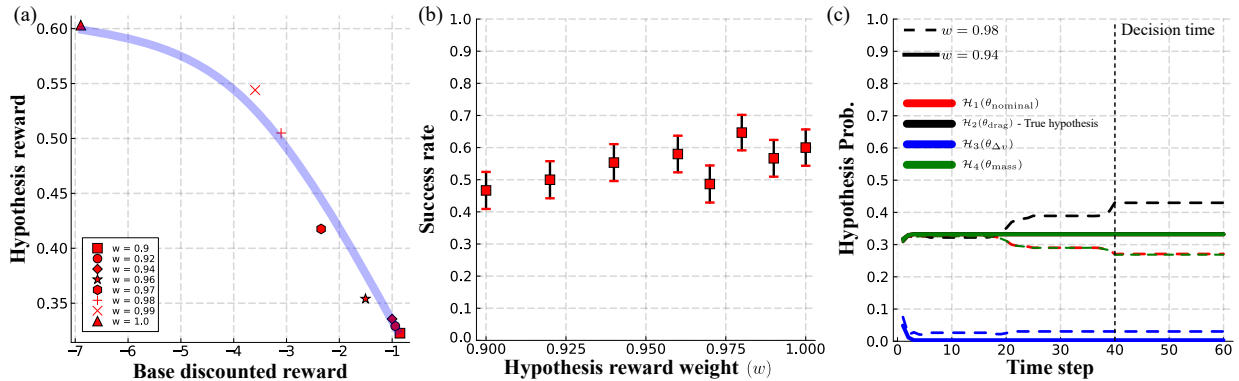


Fig. 4: Pareto front, success rate, and hypothesis probabilities trajectory for the 4-hypotheses example.

and take further observations of the OOI. On the other hand, with the entropy reward, the agent continues to take observations to the OOI after a decision should have been made. For these reasons, the significantly fewer changes to the original plan, and the explicitness of when we want to make a decision, we therefore adopt the commit reward for subsequent experiments.

5.2 Varied Hypotheses and Parameters

This set of experiments focuses on exploring a more varied selection of hypotheses. The scenario is similar to the one described before, only now we add dynamic behaviors that result from a change in mass and engine misfire, modeled as instantaneous velocity addition. We thus have four hypotheses, parameterized by the rv θ_i : \mathcal{H}_1 is the nominal, with drag area of $\theta_1 \sim \mathcal{N}(1, 0.2)$ [m^2] and mass of 8 [kg]; in hypothesis \mathcal{H}_2 the OOI deployed solar panels, with area of $\theta_2 \sim \mathcal{N}(20, 1)$ [m^2], which adds drag; hypothesis \mathcal{H}_3 describes an engine misfire which adds unknown velocity at time 0, $\theta_3 \sim \mathcal{N}(7, 2)$ [m/s]; and lastly hypothesis \mathcal{H}_4 describes a situation where the OOI lowered its mass, so the new mass is $\theta_4 \sim \mathcal{N}(1.33, 0.1)$ [kg]. The true hypothesis is hypothesis 2, with a drag area of $20m^2$. Notice that θ_i for different hypotheses represents a different type of parameter, e.g., in hypotheses 1 – 2 it is the drag area, and in hypothesis 4 it is the OOI’s mass. This gives flexibility in how hypotheses are defined.

The results in Section 5.1 show that for the commit reward, w values smaller than 0.9, do not result in information gathering actions, and leave the base policy unchanged. Figure 4 present average results from 150 MC simulations, where we only considered $w \geq 0.9$. We can see that for the previously chosen weight $w = 0.94$, the performance is not as good as before, with success rate of about 55%. From the Pareto graph we see that for all values of w the hypothesis reward have decreased, even in the limit of $w = 1$. We attribute these results to similar orbits between the hypotheses: while \mathcal{H}_3 , the engine misfire is quickly ruled out as unlikely (Figure 4(c)), it is hard to distinguish between the three other hypotheses by the deadline (red, black, and green lines lie on top of each other). Figure 4(c) shows the mean belief for $w = 0.94$ and $w = 0.98$. It can be seen that for $w = 0.94$ the agent takes, on average, only one observation, at early time steps, where the OOI orbits, under different dynamic models, are still very similar. On the other hand, with $w = 0.98$, the agent takes 8.6 observations on average of the OOI, and we can see that the probability of the correct hypothesis increases. The success rate is then also improved to 65%.

6. SUMMARY

This paper explores the hypothesis-driven sensor tasking problem in SDA scenarios, where the goal is to determine the most accurate hypothesis, while minimally affecting the underlying plan. These types of problems naturally arise in human-machine collaboration, when the human detects a surprising behavior or unexpected outcome and wants to explore it. We formulate the problem as a belief-MDP, dubbed MDH-BMDP, and structure it such that a given POMDP problem can be augmented with different hypotheses that relate to one or more of the problem’s models, and then solved using existing sparse tree search algorithms.

To motivate actions that can help resolve uncertainty and determine the most probable hypothesis while still

performing well in the underlying planning problem, we suggest a new reward function, explicitly rewarding in-time decisions. Simulation results demonstrate the advantage of the new reward function over the entropy-based reward, balancing between timely hypothesis decisions and the underlying problem objectives. The trade-off between the competing objectives can be tuned by the weight w . We explore different values for w and show that for the problem and hypotheses considered in this paper, as we increase w from 0.9 to 1, the agent takes more information gathering actions, observations to the OOI, but at the expense of deviating from the underlying plan. This w gives the human operator a tuning knob, or control, on how much they prioritize determining the most likely hypothesis vs. adhering to the original plan.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under award FA9550-23-1-0726. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFOSR.

7. REFERENCES

- [1] Mauricio Araya, Olivier Buffet, Vincent Thomas, and François Charpillet. A POMDP Extension with Belief-dependent Rewards. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.
- [2] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, March 2012.
- [3] Ryan D. Coder, Charles J. Wetterer, Kris M. Hamada, Marcus J. Holzinger, and Moriba K. Jah. Inferring Active Control Mode of the Hubble Space Telescope Using Unresolved Imagery. *Journal of Guidance, Control, and Dynamics*, 41(1):164–170, January 2018.
- [4] Kyle J. DeMars, Moriba K. Jah, and Paul W. Schumacher. Initial Orbit Determination using Short-Arc Angle and Angle Rate Data. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2628–2637, July 2012.
- [5] Maxim Egorov, Zachary N. Sunberg, Edward Balaban, Tim A. Wheeler, Jayesh K. Gupta, and Mykel J. Kochenderfer. POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty. *Journal of Machine Learning Research*, 18(26):1–5, 2017.
- [6] 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, August 2022.
- [7] Johannes Fischer and Ömer Sahin Tas. Information Particle Filter Tree: An Online Algorithm for POMDPs with Belief-Based Rewards on Continuous Domains. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3177–3187. PMLR, November 2020. ISSN: 2640-3498.
- [8] M. J. Holzinger and M. K. Jah. Challenges and Potential in Space Domain Awareness. *Journal of Guidance, Control, and Dynamics*, 41(1):15–18, January 2018. Publisher: American Institute of Aeronautics and Astronautics.
- [9] A. D. Jaunzemis, M. J. Holzinger, M. W. Chan, and P. P. Shenoy. Evidence gathering for hypothesis resolution using judicial evidential reasoning. *Information Fusion*, 49:26–45, September 2019.
- [10] Andris D. Jaunzemis, Midhun V. Mathew, and Marcus J. Holzinger. Control Cost and Mahalanobis Distance Binary Hypothesis Testing for Spacecraft Maneuver Detection. *Journal of Guidance, Control, and Dynamics*, 39(9):2058–2072, September 2016.
- [11] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 3, pages 1628–1632 vol.3, June 1995.
- [12] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, May 1998.
- [13] Lorena Linares, Rafael Vazquez, Federico Perea, and Jorge Galán-Vioque. A mixed integer linear programming model for resolution of the antenna-satellite scheduling problem. *IEEE Transactions on*

- Aerospace and Electronic Systems*, 60(1):463–473, 2023.
- [14] George Nemhauser and Laurence Wolsey. Computational complexity. *Integer and Combinatorial Optimization*, pages 114–145, 1988.
 - [15] NASA ORBITAL DEBRIS PROGRAM OFFICE. Orbital Debris Quarterly News. *Orbital Debris Quarterly News*, 29(1), February 2025.
 - [16] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, December 2003.
 - [17] Peng Mun Siew and Richard Linares. Optimal Tasking of Ground-Based Sensors for Space Situational Awareness Using Deep Reinforcement Learning. *Sensors*, 22(20):7847, January 2022. Number: 20 Publisher: Multidisciplinary Digital Publishing Institute.
 - [18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018. Publisher: American Association for the Advancement of Science.
 - [19] Zachary Sunberg, Suman Chakravorty, and Richard Scott Erwin. Information Space Receding Horizon Control for Multisensor Tasking Problems. *IEEE Transactions on Cybernetics*, 46(6):1325–1336, June 2016.
 - [20] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, November 2019. Publisher: Nature Publishing Group.
 - [21] William J Wolfe and Stephen E Sorensen. Three scheduling algorithms applied to the earth observing systems domain. *Management Science*, 46(1):148–166, 2000.