(Preprint) AAS 25-726

AUTONOMOUS STRIP IMAGING TASK SCHEDULING IN SUPER-AGILE SATELLITES USING REINFORCEMENT LEARNING

Anaïs Cheval* **and Hanspeter Schaub†**

This paper investigates the use of Deep Reinforcement Learning (DRL) to address the scheduling problem for strip imaging tasks in the context of Super-Agile Earth-Observing Satellites (SAEOS). Unlike point-target imaging, strip imaging enables continuous data collection along extended ground paths, making it essential for monitoring large-scale, elongated features such as international borders, coastlines, and mountain ranges, as well as broad areas decomposed into adjacent strips. A dedicated attitude guidance and control system tailored for strip imaging is developed and integrated into the high-fidelity Basilisk simulation framework. The strip-imaging scheduling problem is modeled as a Partially Observable semi Markov Decision Process (POsMDP), and a custom training environment is built using BSK-RL—a Python package based on Basilisk for constructing Gymnasium environments for spacecraft tasking problems. A DRL based policy is trained and evaluated to allow autonomous on board decision making.

INTRODUCTION

The increasing demand for high-resolution Earth imaging has led to growing complexity in planning missions for Earth-Observing Satellites (EOSs). Applications such as environmental monitoring, disaster response, urban planning, agricultural assessment, and national security drive the need for precise, timely, and efficient data collection across diverse regions. These satellites must balance the execution of imaging tasks with the constraints of system resource limitations. As a result, effective scheduling of mission objectives and onboard resource management is critical to overall mission success. Planning for such satellites is often framed as a flight mode-based problem, where the focus is on selecting which high-level mission objective to pursue or which resource management action to take during a given time period rather than controlling the satellite's actuators directly at a low level.

Traditionally, the ground segment performs the planning and scheduling steps using optimization algorithms. Next the solution is sequenced into commands and up-linked to the spacecraft for open-loop execution. The most popular offline optimization-based approach includes Mixed-Integer Linear Programming (MILP), favored for its optimality guarantees.[1,2] Industry leaders such as Spire Global[3] and Planet[4] employ MILP techniques to manage and schedule their EOS constellations. However, these methods are brittle to initial conditions as they are uploaded to the satellite as an open-loop sequence of flight tasks, present challenges to incorporate non-linear constraints such as wheel torque or power saturation, struggle to scale effectively with a growing number of satellites

*Ph.D. Student, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, 80303. Correspondence: anais.cheval@colorado.edu

†Distinguished Professor and Department Chair, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, 80303. Fellow of AIAA and AAS.

or targets, and require complete or partial re-planing when new requests are introduced into the system.

The use of machine learning has been proposed to overcome these limitations. More specifically, Deep Reinforcement Learning (DRL) has shown the ability to effectively solve the scheduling problem for point imaging tasks under resource constraints for both individual Earth-observing satellites[5–7] and decentralized constellations.[8,9] DRL agents are first trained on high-fidelity simulations to map states to actions to maximize a numerical reward function. A common practice is to train the algorithm using a variety of random initial conditions, targets, and ground station locations to generalize the policy. After the training step, the policy can be up-linked to the satellite for closed-loop execution, responding to the real states of the environment. This means that re-planing is inherent to a DRL planning paradigm. The execution of trained policies is typically very fast. Neural network approximations of the scheduling policy can be executed in milliseconds on modern computational hardware and would not require dedicated evaluation hardware onboard the spacecraft. In some solution, resource management is learned by the policy by directly penalizing failures so that the policy avoids them. In others, a shield (i.e. an expert-designed policy of responses to safety critical states) is deployed during or after learning to guarantee the safety of the satellite during operation.[10,11] Additionally, several studies have conducted in-depth comparisons of different DRL algorithms for this problem,[12] while others have bench-marked DRL against MILP approaches under varying point target distributions.[13]

However, these machine-learning studies have focused exclusively on discrete point-target imaging tasks. This research seeks to build on this previous work by using DRL to address the scheduling problem for continuous strip imaging tasks in the context of Super-Agile Earth Observing Satellites (SAEOSs). Strip imaging tasks are remote sensing operations in which a satellite equipped with a fixed scan line camera captures continuous, high-resolution image data along a defined linear path on the ground as the satellite moves, producing high-resolution images of narrow, extended ground areas. Such imaging is essential for observing features that naturally extend across long distances, including international borders, coastlines, mountain chains, and other large geographic areas that can be decomposed into adjacent strips such as wild-fire areas or flood zones. SAEOSs are especially well-suited for strip imaging tasks thanks to their advanced maneuverability.[14] Leveraging high-performance attitude control systems, these satellites can perform real-time attitude adjustments during an imaging task. This dynamic capability eliminates the traditional constraint of aligning the scanning direction with the satellite's orbital path and decouples the image acquisition rate from the satellite's orbital motion. SAEOSs offer significantly greater flexibility in scheduling strip observations, enabling more efficient and responsive coverage of diverse ground strip targets.

This paper begins by formulating the strip imaging scheduling optimization problem, along with introducing the attitude guidance and control system designed to support such tasks. The scheduling problem is then cast as a Partially Observable semi Markov Decision Process (POsMDP), and the DRL approach employed to solve it is described. Finally, the trained policies are evaluated and the results are discussed.

PROBLEM STATEMENT

Image Strip Request Model

Imaging requests are modeled as a set $\mathcal{R}$ of strip targets, where each request $\rho \in \mathcal{R}$ is defined by a tuple $(\mathbf{r}_{\text{start}}, \mathbf{r}_{\text{end}}, p, v_{\text{acq}})$ representing a start point and end point, both defined as fixed locations in

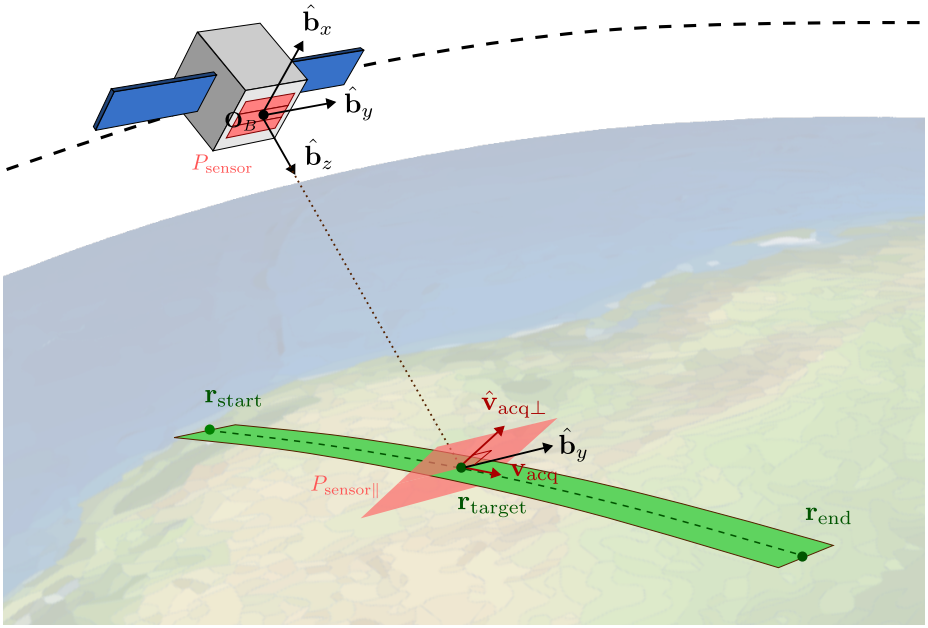

Figure 1: Attitude guidance for a strip imaging task.

the planet-fixed frame, a priority level p, and a required acquisition speed v_{acq}. All imaging requests begin in the unfulfilled set $\mathcal{U}$ and are moved to the fulfilled set $\mathcal{F}$ once successfully imaged. After fulfillment, requests are considered complete and are not re-imaged, although operators may add new requests for the same location if necessary. If a strip can be imaged from both directions, an additional request is generated with the start and end points reversed but sharing the same priority and acquisition speed. These two requests are treated as linked: fulfilling either one causes both to be moved from $\mathcal{U}$ to $\mathcal{F}$.

In this work, synthetic imaging strip requests are generated by first selecting a random start point uniformly distributed over the planet's surface. Each strip is then defined by this start point, an azimuth angle sampled uniformly to determine the direction, and a strip length drawn uniformly from the range $[L_{\text{min}}, L_{\text{max}}]$. The corresponding end point is computed using spherical trigonometry. Each imaging request is assigned a priority value randomly drawn from a uniform distribution over the interval $[0, 1]$, along with an acquisition speed sampled uniformly from the range $[v_{\text{acq,min}}, v_{\text{acq,max}}]$. For each scenario, the total number of imaging requests is randomly sampled from the range $[N_{\text{min}}, N_{\text{max}}]$. Although this specific distribution is used for simulation purposes, the proposed method is general and can accommodate any distribution of strip requests, including those derived from real mission data.

Attitude Guidance And Control Model For A Strip Imaging Task Considering A Super-Agile Satellite

The Earth-observing satellite is modeled as a small spacecraft with mass m and inertia I, operating in a fixed low-altitude circular orbit around Earth with inclination i and altitude a. It is equipped with a body-fixed scan-line camera for imaging purposes. The scan line sensor consists of a rectangular surface containing several rows of photodiodes. Attitude control is provided by a

three-axis reaction wheel assembly, with wheels aligned along the spacecraft's principal body axes. The satellite is super-agile, capable of slewing simultaneously about all three axes—roll, pitch, and yaw—while actively imaging.

Compared to point-imaging tasks, strip-imaging tasks complicate the attitude guidance model by requiring a specific scanning direction at a desired scanning speed. While snapshot instruments permit the orientation around the view direction to remain unspecified during target observation, scan line cameras demand that the scan line sensor aligns perpendicularly with the desired scan path. As a result, the attitude of a satellite equipped with a snapshot instrument retains one unspecified degree of freedom, whereas the attitude must be fully defined for a satellite employing a scan line camera.

To compute the attitude and rate references for strip imaging operations, three primary reference frames are used: the Earth-Centered Inertial frame N, the Earth-Fixed frame E, and the spacecraft Body frame B. The upper-left superscript specifies the reference frame in which the vector is represented, while a lower-right subscript t indicates that the quantity varies with time within that frame. The unit vectors of the body frame $\{\mathbf{O}_B; \hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z\}$ are defined with origin $\mathbf{O}_B$ at the center of the rectangular scan-line sensor, which lies in the plane P_{sensor}. The body frame is visualized in Figure 1, and its unit vectors are defined as follows:

- $\hat{\mathbf{b}}_z$ is the boresight axis of the imager. It is defined as the vector normal to P_{sensor} and passing through $\mathbf{O}_B$.
- $\hat{\mathbf{b}}_y$ is the cross-track axis, defined as the vector lying in P_{sensor}, aligned with the direction of the rows of photodiodes, and passing through $\mathbf{O}_B$. It is oriented to maintain a right-handed body frame.
- $\hat{\mathbf{b}}_x$ is the third body axis, introduced to complete the right-handed orthonormal frame. It is given by
$$\hat{\mathbf{b}}_x = \hat{\mathbf{b}}_y \times \hat{\mathbf{b}}_z,$$
ensuring that $\mathcal{B}$ forms an orthonormal basis.

The primary attitude requirement is to steer the boresight axis $\hat{\mathbf{b}}_z$ towards a virtual ground target ${}^{E}\mathbf{r}_{\text{target},t}$, which moves along the strip's central line — from ${}^{E}\mathbf{r}_{\text{start}}$ to ${}^{E}\mathbf{r}_{\text{end}}$ — with a velocity ${}^{E}\mathbf{v}_{\text{acq},t}$ of constant magnitude v_{acq}. At time t, the positions of the virtual target and the spacecraft in the inertial frame are ${}^{N}\mathbf{r}_{\text{target},t}$ and ${}^{N}\mathbf{r}_{B,t}$, respectively.

The line-of-sight (LOS) vector pointing from the spacecraft to the target is given by:

$$ {}^{N}\mathbf{r}_{LS,t} = {}^{N}\mathbf{r}_{\text{target},t} - {}^{N}\mathbf{r}_{B,t}. \tag{1}$$

The reference boresight direction in the body frame is the normalized transformed LOS vector:

$$ {}^{B}\hat{\mathbf{r}}_{LS,t} = \frac{C_{BN}(t)\,{}^{N}\mathbf{r}_{LS,t}}{\|C_{BN}(t)\,{}^{N}\mathbf{r}_{LS,t}\|}, \tag{2}$$

where $C_{BN}(t)$ is the direction cosine matrix from the inertial frame N to the body frame B, and is derived from the spacecraft's Modified Rodrigues Parameters (MRP) attitude $\boldsymbol{\sigma}_{BN,t}$ at time t.

The principal rotation angle between the boresight axis $\hat{\mathbf{b}}_z$ and ${}^{B}\hat{\mathbf{r}}_{LS,t}$ is:

$$\phi_{1,t} = \arccos\left(\hat{\mathbf{b}}_z \cdot {}^{B}\hat{\mathbf{r}}_{LS,t}\right). \tag{3}$$

The rotation error axis ${}^{B}\hat{\mathbf{e}}_{1,t}$ is defined as:

$${}^{B}\hat{\mathbf{e}}_{1,t} = \begin{cases} \hat{\mathbf{b}}_y, & |\phi_{1,t}| < \epsilon \quad \text{or} \quad |\phi_{1,t} - \pi| < \epsilon, \\ \dfrac{\hat{\mathbf{b}}_z \times {}^{B}\hat{\mathbf{r}}_{LS,t}}{\|\hat{\mathbf{b}}_z \times {}^{B}\hat{\mathbf{r}}_{LS,t}\|}, & \text{otherwise}, \end{cases} \tag{4}$$

where $\epsilon > 0$ is a small threshold introduced to avoid numerical instabilities.

The attitude error between the current body frame B and the desired reference frame $R_1 = \{\mathbf{O}_B; \hat{\mathbf{q}}_x, \hat{\mathbf{q}}_y, \hat{\mathbf{q}}_z\}$ is expressed in MRPs as:

$$\boldsymbol{\sigma}_{BR_1,t} = -\tan\left(\frac{\phi_{1,t}}{4}\right) {}^{B}\hat{\mathbf{e}}_{1,t}, \tag{5}$$

The corresponding reference attitude, which represents the orientation of frame R_1 relative to the inertial frame N, is obtained through MRP composition:

$$\boldsymbol{\sigma}_{R_1N,t} = \boldsymbol{\sigma}_{BN,t} \oplus (-\boldsymbol{\sigma}_{BR_1,t}), \tag{6}$$

where $\oplus$ denotes the MRP addition operation including the shadow set switch to keep the result within the principal MRP domain.

Once the first attitude reference requirement is met, the second requirement ensures that the reference cross track axis aligns perpendicularly with the desired scan path. This is achieved by applying a corrective rotation about the reference boresight axis, resulting in a final attitude reference frame R_2.

To compute the ground track direction at time t, the virtual target velocity vector ${}^{N}\mathbf{v}_{\text{acq},t}$, in the inertial frame, is expressed into the R_1 frame and normalized:

$${}^{R_1}\hat{\mathbf{v}}_{\text{acq},t} = \frac{C_{R_1N}(t)\,{}^{N}\mathbf{v}_{\text{acq},t}}{\left\|C_{R_1N}(t)\,{}^{N}\mathbf{v}_{\text{acq},t}\right\|}, \tag{7}$$

where $C_{R_1N}(t)$ is the direction cosine matrix from the inertial frame N to the reference frame R_1 and is derived from $\boldsymbol{\sigma}_{R_1N,t}$.

To ensure valid scan geometry, the scan path must lie in the sensor plane—orthogonal to the boresight axis—since the perpendicularity between the reference cross-track axis and the scan path is only meaningful when both vectors lie within the same plane. To remove any component of ${}^{R_1}\hat{\mathbf{v}}_{\text{acq},t}$ aligned with the reference boresight axis ${}^{R_1}\hat{\mathbf{r}}_{LS,t}$, this vector is projected onto the plane normal to ${}^{R_1}\hat{\mathbf{r}}_{LS,t}$:

$${}^{R_1}\hat{\mathbf{v}}_{\text{acq}\perp,t} = {}^{R_1}\hat{\mathbf{v}}_{\text{acq},t} - \left({}^{R_1}\hat{\mathbf{v}}_{\text{acq},t} \cdot {}^{R_1}\hat{\mathbf{r}}_{LS,t}\right) {}^{R_1}\hat{\mathbf{r}}_{LS,t}. \tag{8}$$

The reference cross-track axis is given by the normalized vector:

$${}^{R_1}\hat{\mathbf{r}}_{\perp,t} = \begin{cases} {}^{R_1}\hat{\mathbf{v}}_t = \dfrac{{}^{R_1}\hat{\mathbf{r}}_{\text{target},t} \times {}^{R_1}\hat{\mathbf{v}}_{\text{acq},t}}{\left\|{}^{R_1}\hat{\mathbf{r}}_{\text{target},t} \times {}^{R_1}\hat{\mathbf{v}}_{\text{acq},t}\right\|}, & \text{if } \left\|{}^{R_1}\hat{\mathbf{v}}_{\text{acq}\perp,t}\right\| < \epsilon \\ \dfrac{{}^{R_1}\hat{\mathbf{r}}_{LS,t} \times {}^{R_1}\hat{\mathbf{v}}_{\text{acq}\perp,t}}{\left\|{}^{R_1}\hat{\mathbf{r}}_{LS,t} \times {}^{R_1}\hat{\mathbf{v}}_{\text{acq}\perp,t}\right\|}, & \text{otherwise} \end{cases} \tag{9}$$

where ${}^{R_1}\hat{\mathbf{r}}_{\text{target},t} = \frac{C_{R_1N}(t)\,{}^{N}\mathbf{r}_{\text{target},t}}{\left\|C_{R_1N}(t)\,{}^{N}\mathbf{r}_{\text{target},t}\right\|}$, while ${}^{R_1}\hat{\mathbf{v}}_t$ is a unit vector lying in the plane tangent to the Earth's surface at the virtual target location ${}^{R_1}\hat{\mathbf{r}}_{\text{target},t}$ and perpendicular to the velocity vector ${}^{R_1}\hat{\mathbf{v}}_{\text{acq},t}$.

The rotation angle $\phi_{t,2}$ required to align the current cross-track axis $\hat{\mathbf{q}}_y$ with ${}^{R_1}\hat{\mathbf{r}}_{\perp,t}$ is computed as:

$$\phi_{t,2} = \text{sign}\left[-\left(\hat{\mathbf{q}}_y \times {}^{R_1}\hat{\mathbf{r}}_{\perp,t}\right)_z\right] \arccos\left(\hat{\mathbf{q}}_y \cdot {}^{R_1}\hat{\mathbf{r}}_{\perp,t}\right). \tag{10}$$

This corrective rotation about the boresight axis ${}^{R_1}\hat{\mathbf{r}}_{LS,t}$ can be represented as a Modified Rodrigues Parameter (MRP) vector:

$$\boldsymbol{\sigma}_{R_2R_1,t} = -\tan\left(\frac{\phi_{t,2}}{4}\right) {}^{R_1}\hat{\mathbf{r}}_{LS,t}. \tag{11}$$

Finally, the overall reference attitude $\boldsymbol{\sigma}_{R_2N,t}$, expressed relative to the inertial frame N, is obtained by composing the first reference attitude $\boldsymbol{\sigma}_{R_1N,t}$ with this corrective rotation:

$$\boldsymbol{\sigma}_{R_2N,t} = \boldsymbol{\sigma}_{R_1N,t} \oplus \boldsymbol{\sigma}_{R_2R_1,t}. \tag{12}$$

The spacecraft's attitude error relative to this final reference is then:

$$\boldsymbol{\sigma}_{BR_2,t} = \boldsymbol{\sigma}_{BN,t} \ominus \boldsymbol{\sigma}_{R_2N,t}, \tag{13}$$

where $\ominus$ denotes MRP subtraction including the shadow set switch to keep the result within the principal MRP.

The tracking error rate $\dot{\boldsymbol{\sigma}}_{BR_2,t}$ is computed via numerical differentiation of the attitude error $\boldsymbol{\sigma}_{BR_2,t}$ over time. When no prior data point is available, numerical differencing is not feasible. In such cases, the error rate is initialized to zero.

Closed-loop attitude control during a strip imaging task is performed using an exponentially stable MRP-based steering controller,[15] in combination with rate servos driving the three reaction wheels. The reaction wheels are subject to actuation constraints, with commanded torques limited to a maximum u_{max}. The control system operates at a frequency f and receives, at each control step, the attitude error $\boldsymbol{\sigma}_{BR_2,t}$ and the attitude error rate $\dot{\boldsymbol{\sigma}}_{BR_2,t}$.

Imaging Requirements

Imaging requests are subject to operational hard constraints, primarily related to geometric visibility and sensor limitations such as minimum illumination requirements and sensor saturation thresholds. This work focuses on the view angle constraint, which ensures that the virtual target ${}^{N}\mathbf{r}_{\text{target},t}$ is at any time within the sensor's field of regard, but the method is general enough to account for other constraints. Specifically, during a strip imaging task, the spacecraft must maintain:

$$\left|\angle\left({}^{N}\mathbf{r}_{\text{LS},t}, {}^{N}\hat{\mathbf{n}}_t\right)\right| < \frac{\pi}{2} - \theta_{\text{min}} \tag{14}$$

where ${}^{N}\mathbf{r}_{\text{LS},t}$ is the LOS vector, ${}^{N}\hat{\mathbf{n}}_t = \frac{{}^{N}\mathbf{r}_{\text{target},t}}{\|{}^{N}\mathbf{r}_{\text{target},t}\|}$ the local surface normal unit vector at the target and θ_{min} the minimum required elevation angle above the local horizon.

Due to the spacecraft's motion and orbital geometry, only specific time intervals are suitable for initiating the imaging of a strip while satisfying this constraint during the entire task. Starting time opportunity windows are defined as the intervals during which the imaging of a given strip can begin such that the entire strip can subsequently be imaged without violating the view angle constraint. Each of these windows is represented as a time interval $[t_{start,1}, t_{start,2}] = w \in \mathcal{W}_i$, where $\mathcal{W}_i$ denotes the set of all feasible starting time opportunity windows for imaging request i.

In addition to operational hard constraints, dynamic performance requirements must also be met to ensure high-quality imaging. Specifically, during a strip imaging task, the spacecraft's attitude must closely track the guidance profile. This is enforced by bounding during the strip imaging task the attitude error $\boldsymbol{\sigma}_{BR_2,t}$ and its rate $\dot{\boldsymbol{\sigma}}_{BR_2,t}$:

$$\|\boldsymbol{\sigma}_{BR_2,t}\| < \sigma_{\max}, \quad \|\dot{\boldsymbol{\sigma}}_{BR_2,t}\| < \dot{\sigma}_{\max} \tag{15}$$

where $\sigma_{\max}$, $\dot{\sigma}_{\max}$ represent the maximum tolerable attitude error and error rate.

Despite the super-agility of modern satellites, these dynamic performance conditions cannot be guaranteed instantaneously at the start of the imaging interval, as the controller requires a finite transition time to align the spacecraft with the desired attitude and angular velocity. To ensure that the attitude tracking constraints are satisfied from the beginning of the strip, a pre-imaging phase of duration T_{pre} is introduced before each strip imaging task ρ. To incorporate this pre-imaging interval into the simulation process, the central line of the imaging strip is interpolated along the Earth's surface prior to the nominal start location $\mathbf{r}_{\text{start}}$, over a distance equivalent to $v_{\text{acq}} \cdot T_{\text{pre}}$. This interpolation yields a virtual pre-imaging trajectory that precedes the actual imaging path.

Resource Constraints

Earth imaging satellites operate under several resource limitations, including power, onboard memory, and reaction wheel momentum limits. This study focuses solely on the power constraint to isolate and address the core challenge of scheduling strip imaging tasks. The satellite's battery energy, with a maximum capacity $b_{\max}$, must remain positive throughout the entire planning horizon. This is enforced by the constraint:

$$b_t \geq 0 \tag{16}$$

where b_t is the battery energy at time t.

Battery consumption arises from three main sources. First, a constant baseline power consumption, denoted p_{base}, accounts for essential subsystems and is active at all times. Second, the imaging instrument has a power draw of p_{inst} during imaging tasks, provided that imaging requirements are satisfied. Third, the reaction wheels consume electrical power during attitude maneuvers, modeled as $p_{\text{rw}}/\eta_{\text{rw}}$, where p_{rw} is the required mechanical power and $\eta_{\text{rw}} \in (0, 1]$ is the electrical-to-mechanical efficiency.

The satellite is equipped with two solar panels, each with area A, conversion efficiency C_{panel}, and unit normal vector $\hat{n}_{\text{panel}}$. The total electrical power output from these panels is modeled as:

$$p_{\text{out}} = p_{\text{base}} \cdot C_{\text{eclipse}} \cdot C_{\text{panel}} \cdot (\hat{n}_{\text{panel}} \cdot \hat{s}) \cdot A_{\text{panel}} \tag{17}$$

where p_{base} is the incident solar power per unit area at the spacecraft's location, $C_{\text{eclipse}} \in \{0, 1\}$ is the eclipse factor (0 when the spacecraft is in Earth's shadow, 1 when in direct sunlight), $\hat{s}$ is the unit vector pointing from the spacecraft to the sun, and $A_{\text{panel}} = 2A$ is the total panel area.

Passive charging occurs during regular operations when the panels are illuminated by the sun and not in eclipse; in this case, the alignment between the panels and the sun may be suboptimal. Active charging requires reorienting the spacecraft to maximize solar power intake by aligning the panels directly with the sun for a specified duration T_{charge}. The attitude guidance system adopts a similar approach to the one presented to steer the camera's boresight axis toward a virtual target in the strip imaging task. In the active charging scenario, however, the reference attitude is computed to point the body-fixed vector $\hat{n}_{\text{panel}}$ toward the sun's position. To achieve and maintain this orientation, the same MRP-based steering feedback controller used for strip imaging is applied.

Strip Imaging Scheduling Problem

The strip imaging scheduling problem considered in this paper seeks to determine a sequence of actions.

$$\mathbf{a} = (a_1, a_2, \ldots, a_K) \tag{18}$$

where each action a_k is either

- a **charging action** a_{charge} , or
- a **strip imaging action** $a_{\text{strip}} = (\rho, T_{pre})$, where $\rho = (\mathbf{r}_{\text{start}}, \mathbf{r}_{\text{end}}, p, v_{\text{acq}}) \in \mathcal{R}$ is an imaging request and $T_{pre} \in \mathcal{T}_{\text{pre}}$ is a selected pre-imaging time from a set of admissible pre-imaging durations,

such that the cumulative priority of imaging requests that are successfully fulfilled for the first time $\mathcal{F}(\mathbf{a})$ is maximized over the mission duration T.

Formally the problem can be stated as follows :

$$\max_{\mathbf{a}=(a_1,\ldots,a_K)} \quad \sum_{a_{\text{strip}} \in \mathcal{F}(\mathbf{a})} p \tag{19}$$

subject to:

- **Temporal sequencing:** Actions in $\mathbf{a}$ are executed sequentially with start times $t_{0,k}$ satisfying

  $$t_{0,1} = 0, \quad t_{0,k+1} = t_{0,k} + \mathrm{d}(a_k), \quad \forall\, k = 1, \ldots, K-1,$$

  and the entire sequence fits within the mission duration:

  $$t_{0,K} + \mathrm{d}(a_K) \leq T,$$

  where the duration function is defined as

  $$\mathrm{d}(a_k) = \begin{cases} T_{\text{charge}}, & \text{if } a_k = a_{\text{charge}}, \\ T_{pre} + T_{\text{acq}}, & \text{if } a_k = a_{\text{strip}} \end{cases}$$

- **Spacecraft dynamics:** $\forall t \in [0, T],\ \dot{\mathbf{x}}_{\text{sc},t} = f\left(\mathbf{x}_{\text{sc},t},\ \mathbf{u}_{\text{att},t}\right)$
- **Imaging requirements:** $\forall\, a_{strip} \in \mathcal{F}(\mathbf{a}),\ \forall\, t \in [t_0 + T_{pre},\ t_0 + T_{pre} + T_{\text{acq}}]$:

  $$\left|\angle\left({}^{N}\mathbf{r}_{\text{LS},t}, {}^{N}\hat{\mathbf{n}}_t\right)\right| < \frac{\pi}{2} - \theta_{\min}, \quad \|\boldsymbol{\sigma}_{BR_2,t}\| < \sigma_{\max}, \quad \|\dot{\boldsymbol{\sigma}}_{BR_2,t}\| < \dot{\sigma}_{\max}$$

- **Uniqueness of fulfilled requests:** $\forall\, a_{\text{strip}} \in a, \quad |\{k : a_k = a_{\text{strip}} \text{ and } a_k \in \mathcal{F}(\mathbf{a})\}| \leq 1$
- **Resource constraints:** $\forall t \in [0, T],\ b_t \geq 0$

where: $\mathbf{x}_{\text{sc},t}$ denotes the spacecraft state at time t (comprising its translational, attitude and battery states), $\mathbf{u}_{\text{att},t}$ the attitude control input at time t, $f(\cdot)$ the dynamics model provided by a high-fidelity simulator, T_{acq} the time required to image request ρ at an acquisition speed of v_{acq} excluding the pre-imaging phase.

REINFORCEMENT LEARNING

Partially Observable Semi-Markov Decision Processes

The strip imaging scheduling problem is formalized as a Partially Observable semi-Markov Decision Process (POsMDP) to be solved with reinforcement learning. A POsMDP provides a framework for sequential decision-making in environments with partial observability and variable-duration actions. At each decision step, the environment is in a hidden state $s \in \mathcal{S}$, the agent selects an action $a \in \mathcal{A}$, and the environment transitions to a new state $s' \in \mathcal{S}$ according to the transition probability function $T(s' \mid s, a)$. The duration of this transition is governed by the step-duration function $F(s, a, s')$. The agent receives a scalar reward $r = R(s, a, s')$ that quantifies the immediate benefit or cost of taking action a in state s and arriving at state s'. Because the true state s is not directly observable, the agent instead receives an observation $o \in \mathcal{O}$ drawn from the observation function $Z(o \mid s', a)$.

The goal of the RL agent is to learn a policy $\pi(a \mid s)$ that defines the probability of taking action a in state s, aiming to maximize the expected cumulative discounted reward:

$$V(s_0) = \sum_{t=0}^{\infty} \gamma^{\sum_{i=0}^{t} \Delta t_i} r_t \tag{20}$$

where $\gamma \in [0, 1)$ is the discount factor, Δt_i is the duration of step i, r_t is the reward received at decision step t and s_0 denotes the initial state of the system.

POsMDP Formulation For The Strip Imaging Scheduling Problem

The elements of the POsMDP $(\mathcal{S}, \mathcal{A}, T, F, R, \mathcal{O}, Z)$ for the strip imaging scheduling problem are defined as follows :

1. **State space** $\mathcal{S}$ is the complete space of simulator states required to maintain the Markov assumption. It includes satellite dynamic states, flight software states, and environment states. A terminal state is reached at step k if there exists a time t during step k such that $b_t = 0$, indicating the satellite can no longer operate.
2. **Action space** $\mathcal{A}$ is composed of 2 modes :
    - **Imaging:** To avoid an excessively large action space, not all imaging requests are included in the decision process at each step. Instead, only the next N unfulfilled requests illustrated on Figure 2 are considered. These are ordered based on the remaining time until the closing of their respective starting time opportunity windows. At each decision

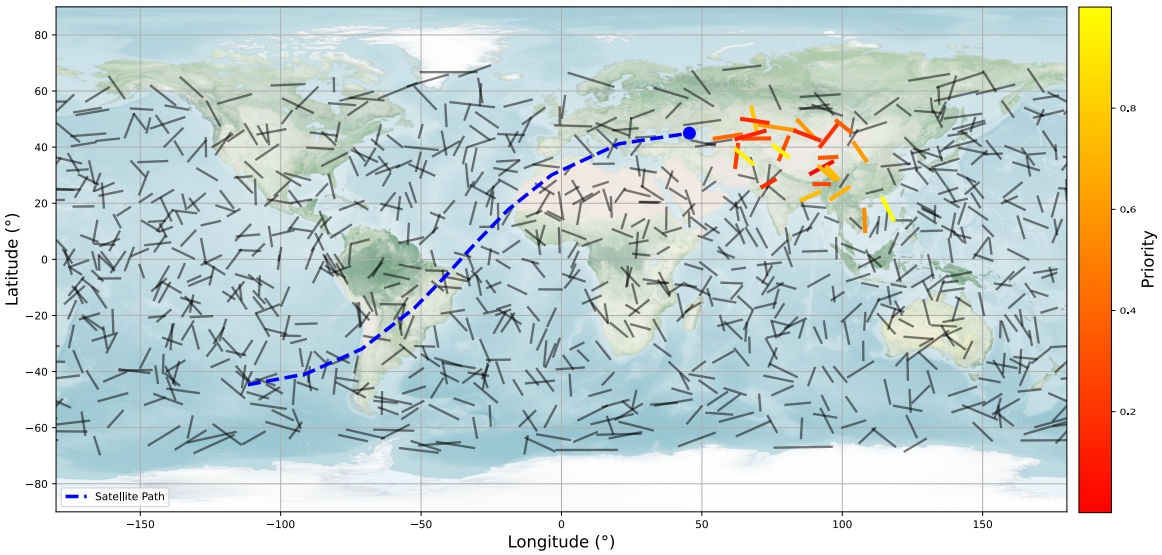

Figure 2: Action space for imaging tasks composed of the next N unfulfilled strips.

step, the agent selects a tuple $(r, T_{\text{pre}}) \in [1, N] \times \mathcal{T}_{\text{pre}}$, where r denotes a request index among the selected N unfulfilled strips, and T_{pre} is a pre-imaging time chosen from a discrete set $\mathcal{T}_{\text{pre}}$ of allowable values.

- **Charging:** The satellite points its solar panels toward the Sun, turns off all instruments and recharges its batteries. The action duration is set to T_{charge}.

Formally, the action space is defined as:

$$\mathcal{A} = \left([1, N] \times \mathcal{T}_{\text{pre}}\right) \cup \left\{a_{\text{charge}}\right\}.$$

3. **Transition probability function** T is deterministic and defined by a generative model G such that $G(s, a)$ returns the next state s'. Then,

$$T(s, a, s') = \begin{cases} 1 & \text{if } s' = G(s, a) \\ 0 & \text{otherwise} \end{cases}$$

4. **Step-duration function** F is deterministic and depends solely on the chosen action a. Specifically, F maps each action to its associated execution time, as defined by:

$$F(a) = \begin{cases} T_{\text{pre}} + T_{\text{acq,r}} & \text{if } a = (r, T_{\text{pre}}) \in [1, N] \times \mathcal{T}_{\text{pre}} \\ T_{\text{charge}} & \text{if } a = a_{\text{charge}}. \end{cases}$$

5. **Reward function** R yields the priority of the request if it is fulfilled for the first time, and zero otherwise:

$$R(s, a, s') = \begin{cases} p_r & \text{if } a = (r, T_{\text{pre}}),\ \rho_r \in \mathcal{U}(s),\ \text{and } \rho_r \in \mathcal{F}(s') \\ 0 & \text{otherwise.} \end{cases}$$

Table 1: Observation Space

Parameter	Normalization	Dimension	Description
b	b_{max}	1	Battery energy normalized by maximum battery capacity b_{max}
Φ_s	$\pi\,\text{rad}$	1	Angle between solar panels and sun vector
t^o_{Ecl}	T	1	Time until the next eclipse starts, normalized by the orbital period T
t^c_{Ecl}	T	1	Time until the next eclipse ends, normalized by the orbital period T
$p_{n\in N}$	-	N	Priority of next N unfulfilled requests
$l_{n\in N}$	L_{max}	N	Length of next N unfulfilled requests
$v_{\text{acq},n\in N}$	$v_{\text{acq,max}}$	N	Required acquisition speed for the next N unfulfilled requests
$\boldsymbol{\theta}_{BR_2,\,n\in\mathbb{N}}$	$\pi\,\text{rad}$	N	Attitude error angle between B and R_2 if the spacecraft starts imaging without pre-imaging for the next N unfulfilled requests
$\dot{\boldsymbol{\theta}}_{BR_2,\,n\in\mathbb{N}}$	$0.04\,\text{rad/s}$	N	Attitude rate error between B and R_2 if the spacecraft starts imaging without pre-imaging for the next N unfulfilled requests
$w_{\text{relative},n\in N}$	$300\,\text{s}$	$2N$	Next starting time opportunity window for the next N unfulfilled requests expressed relative to the current simulation time

6. **Observation space** $\mathcal{O}$ detailed in Table 1 is constructed by selecting and transforming relevant dimensions from the full state space, guided by expert knowledge and ablation studies. It includes key information about the spacecraft and the next N upcoming unfulfilled requests. These observations are limited to data that the satellite can reasonably obtain onboard with minimal uncertainty, supporting reliable closed-loop decision-making. All observation elements are normalized to approximately lie within the range $[-1, 1]$ to enhance the performance of RL algorithms.

7. **Observation function** Z is deterministic since the satellite is assumed to observe the observation space perfectly.

The POsMDP and underlying generative model are implemented using BSK-RL[*], a modular, open-source package for creating spacecraft tasking RL environments. BSK-RL uses the standard Gymnasium API for RL environments, making the package compatible with all major RL frameworks. Internally, the spacecraft and environment dynamics are modeled using the Basilisk[†] spacecraft simulation framework.

Learning On An Infinite Horizon POsMDP with RL

RL is employed to approximate solutions to POsMDPs, using the widely adopted Proximal Policy Optimization (PPO) algorithm. PPO is a stochastic policy-gradient method that updates the

[*] `https://github.com/AVSLab/bsk_rl`
[†] `https://github.com/AVSLab/basilisk`

policy π_θ through iterative improvements to the network parameters θ as the agent interacts with the environment. To maintain stability, PPO constrains updates to stay within a small region around the current policy. Given a batch of trajectories collected under parameters θ_k, the updated parameters θ_{k+1} are computed by maximizing the clipped surrogate objective:

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_t \left[\min \left(l_t(\theta)\, \hat{A}_t,\ \text{clip}\left(\rho_t(\theta),\, 1-\epsilon,\, 1+\epsilon\right)\, \hat{A}_t \right) \right], \tag{21}$$

where $l_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$ is the probability ratio between the new and old policy, $\hat{A}_t$ is the estimated advantage function at time t (GAE), and ϵ is a small hyperparameter that bounds the range of policy updates.

In the strip imaging scheduling problem, actions occur over variable time intervals and an infinite time horizon is considered, as the mission duration T is assumed to be large. PPO is implemented using RLlib. The infinite-horizon formulation is handled by truncating episodes and using RLlib's default bootstrapping mechanism to estimate the value function beyond the truncation point. However, RLlib does not natively support variable time intervals in its advantage estimation. To address this, we adapt the GAE to account for elapsed time between decisions, using the value function $V(s_t)$ defined previously for the semi-Markov setting [Eq. 20] :

$$\hat{A}_t = \sum_{i=0}^{\infty} (\lambda\gamma)^i \sum_{j=0}^{i} \Delta_{t+j} \left(r_{t+i} + \gamma^{\Delta_{t+i+1}} V(s_{t+i+1}) - V(s_{t+i}) \right), \tag{22}$$

where $\lambda \in [0,1]$ is the GAE's decay parameter, $\gamma \in [0,1]$ is the discount factor, Δ_{t+j} is the time duration between steps $t+j$ and $t+j+1$, r_{t+i} is the reward at time $t+i$, and $V(s_{t+i})$ is the semi-Markov value function at state s_{t+i}.

This formulation ensures that rewards and value estimates are discounted according to the actual time elapsed rather than the number of steps, making PPO applicable to variable-time decision processes.

Shielding For RL

Shielded RL, originally proposed by Alshiekh et al. in Reference 16, enhances the safety of RL agents by incorporating a decision-making mechanism called a shield, which enforces formal safety guarantees during policy execution. Two primary shielding mechanisms exist: one that is integrated directly into the training process and another applied post-training. In this work, we adopt the latter post-processing approach, where the shield monitors each action selected by the trained policy based on the current observation, allowing safe actions to proceed while replacing unsafe ones with predefined safe alternatives. This choice is motivated by prior findings[17] suggesting that separating the shield from the training process often results in improved interaction between the learned policy and the shielding system, as the policy is trained freely in an unconstrained environment. In this work, safety considerations arise due to the limited onboard battery energy. To address this, a simple, hand-crafted shield is employed, which selects the charging action whenever the battery level falls below a predefined threshold, denoted z_{minsafe}, defined as follows:

$$z_{\text{minsafe}} = \begin{cases} z_{\text{floor}} - (t^c_{\text{Ecl}} - t^o_{\text{Ecl}})\dot{z}_{\text{draw}} - t^o_{\text{Ecl}}\dot{z}_{\text{gain}} & \text{if not in eclipse and } z > z_{\text{floor}} \\ z_{\text{floor}} - t^c_{\text{Ecl}}\dot{z}_{\text{draw}} & \text{if in eclipse} \\ z_{\text{floor}} & \text{else} \end{cases} \tag{23}$$

This expression maintains the battery level above z_{floor}, a lower bound considered sufficient to perform any imaging task. A higher threshold is enforced as an eclipse approaches to ensure that the agent can survive the eclipse duration without depleting its energy. The term $\dot{z}_{\text{gain}}$ represents the passive charging rate of the satellite during sunlight, while $\dot{z}_{\text{draw}}$ denotes the estimated energy consumption rate of the satellite.

Table 2: Simulation Parameters

Parameter	Value	Parameter	Value
Spacecraft Properties		**Steering Controller**	
(a, i, e)	$(600\,\text{km}, 45°, 0)$	$(K_1, K_3, \omega_{\text{max}})$	$(3,\ 10,\ 5\,\text{rad}\,\text{s}^{-1})$
Other Orbital Parameters	Randomized	**Imaging Requirements**	
m	$330\,\text{kg}$	σ_{max}	0.05
I	$[121,\ 98,\ 82]\,\text{kg}\,\text{m}^2$	$\dot{\sigma}_{\text{max}}$	$0.1\,\text{rad}\,\text{s}^{-1}$
b_{max}	$1.44 \times 10^6\,\text{W}\,\text{s}$	θ_{min}	$10°$
b_0	$[0.4,\ 0.6]\,b_{\text{max}}$	**Request Properties**	
p_{base}	$1\,\text{W}$	$[L_{\text{min}}, L_{\text{max}}]$	$[500, 1000]\,\text{km}$
p_{inst}	$20\,\text{W}$	$[v_{\text{acq,min}}, v_{\text{acq,max}}]$	$[2, 4]\,\text{km}\,\text{s}^{-1}$
η_{rw}	0.5	$[N_{\text{min}}, N_{\text{max}}]$	$[1000, 2000]$
C_{panel}	0.2	**Shield Properties**	
A_{panel}	$2\,\text{m}^2$	z_{floor}	$0.2\,b_{\text{max}}$
$\hat{n}_{\text{panel}}$	$[0,\ 1,\ 0]$	$\dot{z}_{\text{draw}}$	b_{max} per orbit
T_{charge}	$5\,\text{min}$	$\dot{z}_{\text{gain}}$	$0.5\,b_{\text{max}}$ per orbit
u_{max}	$1\,\text{N}\,\text{m}$		

RESULTS

Training Environment

The training environment is configured using the simulation parameters listed in Table 2 and training hyperparameters in Table 3 . Unlisted values default to standard settings in BSK-RL and RLlib v2.6.3. Training was conducted on the University of Colorado's Research Computing (CURC) infrastructure, using 32 cores and up to 20M steps. In total, three different policies were trained, each using a distinct discrete set of allowable pre-imaging time values, denoted as $\mathcal{T}_{\text{pre}}$. Policy π_1 was restricted to a single value, with $\mathcal{T}_{\text{pre}} = \{60\}$ seconds. Policy π_2 was allowed to choose between two options, with $\mathcal{T}_{\text{pre}} = \{10, 60\}$ seconds. Policy π_3 was trained with the most flexibility, selecting from $\mathcal{T}_{\text{pre}} = \{10, 35, 60\}$ seconds. This incremental expansion in the action space was designed to assess the impact of pre-imaging timing flexibility on agent performance and overall imaging efficiency. All policies were trained without any shield mechanisms, meaning agents were required to learn avoiding low battery states without external constraints during training. Following training, a safety shield was applied only to the best-performing policy to enforce operational safety guarantees.

Table 3: Training Hyperparameters

Parameter	Value
Neural network	2 layers with 2048 neurons each
Number of workers	32
Learning rate	3.10^{-5}
Discount factor	0.9999
Training batch size	3000
Number of SGD iterations	10
GAE's decay parameter λ	0.95
Gradient clipping	0.2
PPO clipping parameter	0.5
N	25

In Distribution Performance Evaluation

The in-distribution performance of the trained policies is evaluated, meaning the same parameters as those applied during training are used. In particular, the evaluation time horizon is fixed to five orbits, consistent with the duration of the training episodes.

To analyze the impact of request density on policy performance, the total number of imaging requests, denoted $|\mathcal{R}|$, is varied across five values: 1000, 1250, 1500, 1750, and 2000. During training, $|\mathcal{R}|$ is randomly sampled within the range $[1000, 2000]$ for each episode to ensure that the learned policy generalizes across different request densities. For each combination of policy π and request set size $|\mathcal{R}|$, 200 test cases are executed to ensure statistically reliable comparisons. The average cumulative reward, along with the average values of relevant characteristics of the selected strips—namely acquisition duration (given by $l \cdot v_{\text{acq}}$, and not including T_{pre}), target priority, and attitude angular error with the strip's starting point before starting the task—are reported in Figure 3. As a baseline, the average values of these strip characteristics are also shown assuming the policy selects strips randomly.

For all policies, the average cumulative reward consistently increases as the request density grows. This behavior aligns with expectations as a higher density provides a broader set of requests that satisfy the view-angle constraint. With more options available, the policy can become increasingly selective. However, the way this selectivity is expressed varies across policies, depending on the structure of the available pre-imaging time set $\mathcal{T}_{\text{pre}}$. Specifically:

- Policy π_1, restricted to a fixed 60 seconds pre-imaging duration, adapts to increasing request density by prioritizing strips with higher target value and shorter acquisition times. While it cannot shorten pre-imaging, it reduces total imaging time by selecting shorter acquisitions—allowing more strips to be imaged. The average angular error remains lower than that of the random policy, as strips requiring excessive slews are avoided. However, this error does not decrease with higher request density, indicating the policy prevents maneuvers that exceed the 60-second limit but gains no advantage from further reducing angular error once the fixed pre-imaging time allows completion of the necessary slew.

- Policy π_2, which allows two pre-imaging durations (10 and 60 seconds), exhibits a different strategy. Like π_1, it prioritizes strips with higher target values. However, it also reduces the

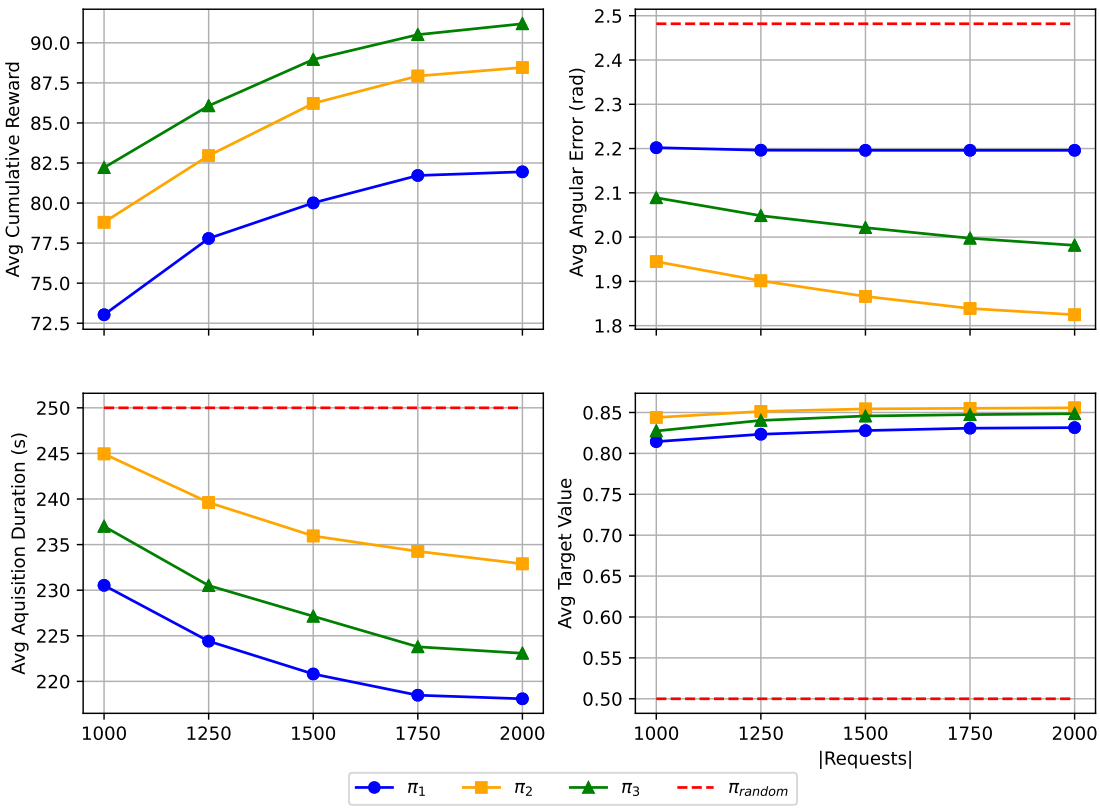

Figure 3: Impact of request density on π_1,π_2,π_3.

average angular error as request density increases. This indicates an effort to enable the use of the shorter 10 seconds pre-imaging option when feasible, even at the cost of selecting strips with slightly longer acquisition durations.

- Policy π_3, which offers the most flexibility with three pre-imaging durations (10, 35, and 60 seconds), adopts a more balanced strategy. It simultaneously improves target value and reduces both acquisition duration and angular error. This enables π_3 to reduce total imaging time through a combination of shorter acquisition tasks and more efficient slews.

All policies tend to favor strips with higher target values and aim to reduce the total duration of imaging tasks—either by shortening the acquisition phase, the pre-imaging phase, or both. This reflects the agent's awareness—enabled by the semi-Markov formulation—that each task has a variable duration, and that reducing this duration allows more strips to be scheduled within the available time window.

To further investigate policy behavior and how they accommodate imaging and charging constraints, 200 evaluation episodes are run per policy, each with a random number of imaging requests within the range $[1000, 2000]$. The average episode time share across activities is reported in Figure 4. Policy π_2 exhibits the lowest average pre-imaging duration per strip (25.0 s/strip) for success-

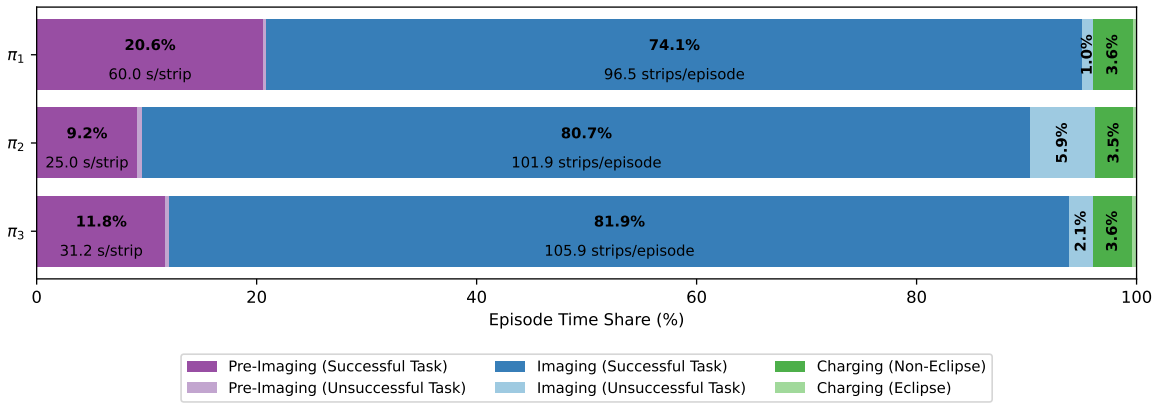

Figure 4: Average episode time share for π_1,π_2,π_3.

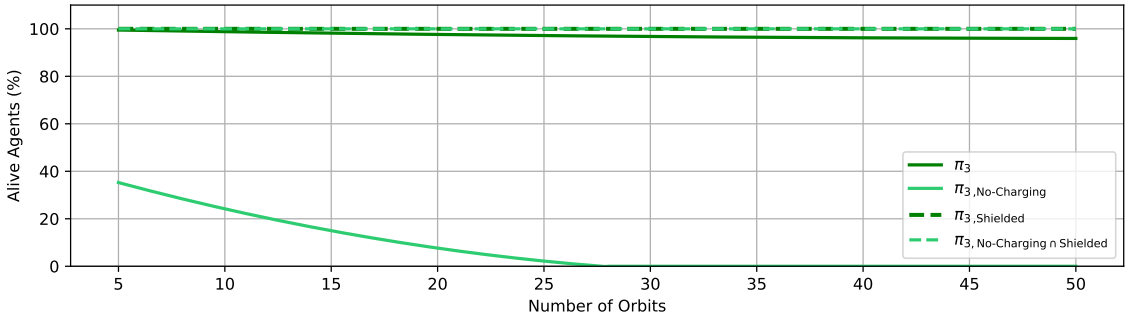

Figure 5: Survival rate per orbit for π_3 and its variants.

ful tasks, indicating frequent use of the shorter 10 seconds option. However, this aggressive strategy leads to the highest share of failed tasks—5.9% of total episode time, compared to only 2.1% for π_3 and 1% for π_1 . While policy π_2 wastes time due to frequent failed attempts resulting from overly short pre-imaging durations, policy π_1, which uses a fixed 60 seconds pre-imaging window, tends to over-allocate slewing time. This results in the lowest time share for successful imaging—74.1% of total episode time, compared to 80.7% for π_2 and 81.9% for π_3. Policy π_3 has a similar successful imaging share as policy π_2, achieving this by reallocating time lost to failures into longer, more reliable pre-imaging phases. However, despite the similar imaging time share, policy π_3 completes significantly more strips per episode (105.9 vs. 101.9 for π_2) by focusing more on reducing acquisition duration, thereby using the available imaging time more efficiently. All policies exhibit comparable charging behavior, accounting for approximately 3.5% of the total episode time with minimal eclipse-related charging time , while achieving a probability of survival exceeding 99.5% on 5 orbits.

Overall, policy π_3, with its balanced strategy, performs best—attaining the highest average cumulative reward across all request densities, as shown in Figure 3. This outcome underscores the advantages of greater pre-imaging flexibility in supporting effective scheduling decisions.

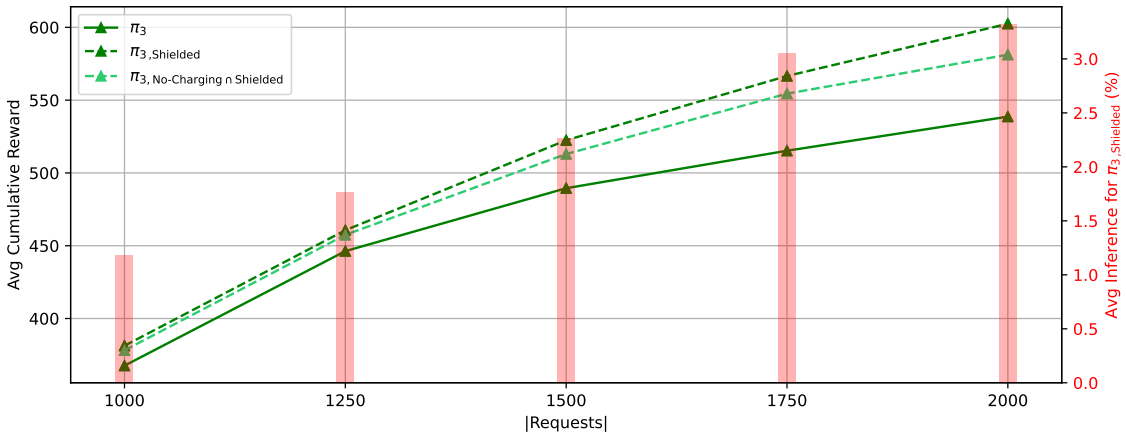

Figure 6: Cumulative reward for π_3, $\pi_{3,\text{No-Charging} \cap \text{Shielded}}$ and $\pi_{3,\text{Shielded}}$ with corresponding shield inference.

Out of Distribution Performance Evaluation

The in-distribution performance evaluation highlighted policy π_3 as the best-performing strategy achieving a survival probability exceeding 99.5% over 5 orbits. To assess its safety beyond those 5 orbits, out-of-distribution testing is performed by extending the evaluation time horizon.

Policies were trained on 5-orbit episodes, using RLlib's bootstrapping mechanism at truncation points to simulate infinite-horizon learning. The ultimate objective is to obtain a safe policy suitable for deployment over long-duration missions. To analyze the impact of the time horizon on survival rate, tests are conducted using 50-orbit-long episodes—ten times longer than those used during training. A total of 200 such episodes are simulated, and the survival rate per orbit is reported in Figure 5. Policy π_3 is compared against three variants: $\pi_{3,\text{Shielded}}$, a post-training shielded version of the same policy using the shield defined in Equation (23); $\pi_{3,\text{No-Charging}}$, a policy trained under the same conditions as π_3 but assuming infinite battery capacity; and $\pi_{3,\text{No-Charging} \cap \text{Shielded}}$, which combines $\pi_{3,\text{No-Charging}}$ with the post-processing shield. Since $\pi_{3,\text{No-Charging}}$ does not learn active charging behaviors and relies solely on passive charging, it serves as a baseline for assessing the influence of battery constraints on both safety and performance. After 50 orbits, the survival rate of policy π_3 drops to 92.5%, while the shielded versions maintain both a survival rate of 100%. In contrast, $\pi_{3,\text{No-Charging}}$ results in complete failure, with a 0% survival rate.

To compare the performance of π_3, $\pi_{3,\text{Shielded}}$, and $\pi_{3,\text{No-Charging} \cap \text{Shielded}}$, 200 experiments are conducted for each of five request values: 1000, 1250, 1500, 1750, and 2000. As shown in Figure 6, $\pi_{3,\text{Shielded}}$ consistently outperforms both alternatives. Compared to the base policy π_3, $\pi_{3,\text{Shielded}}$ performs better primarily because it stays alive longer during long-horizon episodes. While slightly more conservative—intervening in approximately 2.26% of actions by replacing imaging with charging when safety thresholds are at risk—the post-processing shield prevents critical failures. These interventions allow the policy to remain operational and accumulate more reward over time. Compared to $\pi_{3,\text{No-Charging} \cap \text{Shielded}}$, $\pi_{3,\text{Shielded}}$ also achieves higher performance, especially under high imaging request densities. This suggests that when the policy is trained with safety constraints, it learns to select better times to charge.

CONCLUSION

This paper presents a guidance and control model for strip imaging tasks and explores the use of DRL to solve the strip imaging scheduling problem. The results show that policies prioritize strips with high target values and seek to reduce the total imaging duration per strip by shortening the acquisition and/or pre-imaging phases to enable imaging of more strips. Policies with finer granularity in pre-imaging time choices achieve the best performance, resulting in higher cumulative rewards. Regarding safety, unshielded policies offer no guarantee against mission loss. The safety shield effectively prevents critical issues, enabling sustained operation. This suggests that agents can be trained without safety constraints and later deployed with shields to ensure survivability. However, policies trained with safety awareness achieve better performance, particularly in scenarios with higher target densities. Future work will extend the framework to consider partial imaging of strips and incorporate uncertainties such as cloud cover.

ACKNOWLEDGMENT

Partial support for this work was received through a Phase I Space Force SBIR grant with No. FA2541-24-C- B044.

REFERENCES

[1] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers & Operations Research*, Vol. 57, 2015, pp. 1–16.

[2] D. Eddy and M. Kochenderfer, "Markov decision processes for multi-objective satellite task planning," *2020 IEEE Aerospace Conference*, IEEE, 2020, pp. 1–12.

[3] J. Cappaert, F. Foston, P. S. Heras, B. King, N. Pascucci, J. Reilly, C. Brown, J. Pitzo, and M. Tallhamm, "Constellation modelling, performance prediction and operations management for the spire constellation," 2021.

[4] V. Shah, V. Vittaldev, L. Stepan, and C. Foster, "Scheduling the world's largest earth-observing fleet of medium-resolution imaging satellites," *International Workshop on Planning and Scheduling for Space*, Organization for the 2019 International Workshop on Planning and Scheduling, 2019, pp. 156–161.

[5] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, and G. Wu, "A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 3, 2020, pp. 1463–1474.

[6] A. Harris, T. Valade, T. Teil, and H. Schaub, "Generation of Spacecraft Operations Procedures Using Deep Reinforcement Learning," *Journal of Spacecraft and Rockets*, Vol. 59, March – April 2022, pp. 611–626, 10.2514/1.A35169.

[7] A. Herrmann and H. Schaub, "Reinforcement learning for the agile earth-observing satellite scheduling problem," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 5, 2023, pp. 5235–5247.

[8] A. Herrmann, J. a. V. Carneiro, and H. Schaub, "Reinforcement Learning for The Multi-Satellite Earth-Observing Scheduling Problem," *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, Springer, 2022, pp. 1351–1368.

[9] A. Herrmann, M. A. Stephenson, and H. Schaub, "Single-Agent Reinforcement Learning for Scalable Earth-Observing Satellite Constellation Operations," *Journal of Spacecraft and Rockets*, Vol. 61, No. 1, 2024, pp. 114–132.

[10] A. Harris and H. Schaub, "Spacecraft Command and Control with Safety Guarantees Using Shielded Deep Reinforcement Learning," *AIAA SciTech*, Orlando, Florida, Jan. 2020, pp. Jan. 6–10.

[11] I. Nazmy, A. Harris, M. Lahijanian, and H. Schaub, "Shielded Deep Reinforcement Learning for Multi-Sensor Spacecraft Imaging," *American Control Conference*, Atlanta, Georgia, June 2022, pp. June 8–10.

[12] A. Herrmann and H. Schaub, "A comparative analysis of reinforcement learning algorithms for earth-observing satellite scheduling," *Frontiers in Space Technologies*, Vol. 4, 2023, p. 1263489.

[13] M. Stephenson, L. Mantovani, A. Cheval, and H. Schaub, "Quantifying the Optimality of a Distributed Rl-Based Autonomous Earth-Observing Constellation,"

[14] X. Wang, G. Wu, L. Xing, and W. Pedrycz, "Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions," *IEEE Systems Journal*, Vol. 15, No. 3, 2020, pp. 3881–3892.

[15] H. Schaub and S. Piggott, "Speed-constrained three-axes attitude control using kinematic steering," *Acta Astronautica*, Vol. 147, 2018, pp. 1–8.

[16] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32, 2018.

[17] R. Reed, H. Schaub, and M. Lahijanian, "Shielded deep reinforcement learning for complex spacecraft tasking," *2024 American Control Conference (ACC)*, IEEE, 2024, pp. 2331–2337.