Reinforcement learning is used to design reconfiguration maneuvers for a spacecraft to adjust its position relative to another spacecraft located along an \( L_2 \) halo orbit in the Sun-Earth circular restricted three-body problem. This specific scenario is modeled after a starshade reconfiguring to block starlight while a space telescope observes exoplanets. First, reconfiguration maneuver design is translated into a reinforcement learning problem. Then, Proximal Policy Optimization is used to train a policy that generates sequences of impulsive reconfiguration maneuvers. The trained policy is examined and used to produce reconfiguration maneuver sequences that successfully achieve the reconfiguration goals with low maneuver magnitudes.

1 INTRODUCTION

Autonomous maneuver design will likely become a key asset for creating intelligent and resilient spacecraft that operate in multi-body systems. Among the variety of maneuvers a spacecraft may perform during a mission, reconfiguration maneuvers are used to guide either a cluster or a single spacecraft to a new relative geometric configuration or reference path.\(^1\)\(^,\)\(^2\) Existing approaches for designing reconfiguration maneuvers in multi-body systems use a combination of tools from dynamical systems theory and optimization.\(^3\)\(^-\)\(^5\) Although these strategies are successful, they often require significant computational resources as well as involvement of a human analyst on the ground. These requirements may be prohibitive when a rapid response is needed or when communications are limited. Strategies for autonomous reconfiguration maneuver design may address these challenges, reduce operational support requirements, and eventually enable a variety of future missions including constellations, large structures assembled on-orbit, and small satellites.

Reinforcement Learning (RL) offers one potential approach to autonomous maneuver design in multi-body systems. An RL implementation is composed of several fundamental components: an agent, an environment, a state vector, an action to perform for that state, and a reward function that is evaluated for each state-action pair.\(^6\) An RL algorithm is then used to train a policy, mapping a

---

\(^1\)Assistant Professor, Colorado Center for Astrodynamics Research (CCAR), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303  
\(^2\)Graduate Researcher, Colorado Center for Astrodynamics Research (CCAR), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303  
\(^3\)Associate Professor, Colorado Center for Astrodynamics Research (CCAR), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303  
\(^4\)Associate Professor, Research and Engineering Center for Unmanned Vehicles (RECUV), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303
state to an action, to maximize the expected total reward within an environment.\textsuperscript{7} The capability of RL-based implementations to recover optimal policies for trajectory design and guidance in multi-body systems has recently been explored by several researchers. For instance, Miller and Linares,\textsuperscript{8} Sullivan and Bosanac,\textsuperscript{9–11} and LaFarge et al.\textsuperscript{12} use RL algorithms to design low-thrust enabled orbit transfers in chaotic environments. Moreover, Scorsoglio et al. apply an RL algorithm to the problem of relative motion around periodic orbits,\textsuperscript{13} while Molnar\textsuperscript{14} and Bonasera et al.\textsuperscript{15} explore RL-based implementations for autonomously designing station-keeping maneuvers for spacecraft operating near periodic orbits in multi-body systems.

To explore the application of RL to autonomous maneuver design in one type of reconfiguration problem, consider a starshade operating in the vicinity of a space telescope. The starshade must maneuver during a transfer phase to reconfigure its position relative to the space telescope. The starshade must also approximately maintain that relative position over a specified duration to block starlight while the space telescope observes an exoplanet. These reconfiguration maneuvers occur frequently throughout the mission with various durations for each of the transfer and observation periods as the telescope observes a variety of exoplanets. Such a scenario has been explored over the years for a variety of concepts and missions involving spacecraft operating near libration points in the Sun-Earth system. For instance, the Nancy Grace Roman Space Telescope, Exo-S and the New Worlds Observer have each considered a starshade operating near a space telescope in an orbit near Sun-Earth $L_2$.\textsuperscript{16–18} As a result, there is extensive literature available to inform formulating a simplified scenario for autonomous reconfiguration maneuver design. Furthermore, researchers including Webster and Folta, Farré\’s and Webster, Kolemen and Kasdin, Flinois et al. and Soto et al. have previously examined the maneuvers required to regularly reconfigure a starshade to possess a desired position vector relative to a space telescope.\textsuperscript{17, 19–22} The availability of this literature facilitates a valuable high-level comparison to the results generated in this paper.

In this paper, RL is used to autonomously design impulsive reconfiguration maneuvers for a starshade operating in the Sun-Earth circular restricted three-body problem (CR3BP). Specifically, reconfiguration maneuver sequences are designed to modify the position vector of a starshade relative to a space telescope that is located along a halo orbit near Sun-Earth $L_2$. Using RL to solve this problem enables the construction of a policy that maps a variety of initial and desired relative configurations as well as durations for the target and observation phases to the maneuver sequences that balance the relative positioning goals with minimizing control effort. This paper first presents the translation of this reconfiguration maneuver design scenario into an RL problem. Then, due to its observed efficacy and convergence properties for training locally optimal policies in multi-body systems, Proximal Policy Optimization (PPO) is used in an actor-critic configuration to train a policy to generate the reconfiguration maneuvers.\textsuperscript{8–12, 15} The trained policy is then evaluated to generate reconfiguration maneuver sequences for the starshade. This policy is examined in the context of two evaluation scenarios: 1) across single maneuver sequences associated with a variety of relative positioning goals, initial configurations, and combinations of transfer and observation phase durations; and 2) across successive maneuver sequences to frequently reconfigure the starshade and target a list of desired relative position vectors for specified observation phase durations over the span of 200 days.

2 BACKGROUND: DYNAMICAL MODEL

The CR3BP is used in this paper to approximate the motion of a spacecraft in the Sun-Earth system. This particular model is used in this preliminary implementation of an RL-based reconfig-
uration maneuver planner to balance sufficiently representing the complex dynamical environment with the computational effort required to numerically integrate a large number of trajectories during training. In the CR3BP, the motion of a spacecraft of negligible mass is modeled under the gravitational influences of two more massive primary bodies, i.e., the Sun and the Earth. The primaries are denoted as $P_1$ and $P_2$ with masses $M_1$ and $M_2$, respectively, and are assumed to follow circular orbits about the mutual barycenter. Then, a rotating frame $(\hat{x}, \hat{y}, \hat{z})$, is defined to rotate with the primaries: the $\hat{x}$-axis is directed from $P_1$ to $P_2$, the $\hat{z}$-axis is aligned with the orbital angular momentum vector of the primary system, and the $\hat{y}$-axis is defined to complete the right-handed triad. Neural networks are initialized with no knowledge of the environment and initially act randomly. Then, the agents that gather state-action-reward experiences in the environment. 

Next, state quantities are nondimensionalized using the characteristic quantities $l^*, t^*$, and $m^*$ to reduce the potential for ill-conditioning: $l^*$ is defined as the assumed constant distance between the primaries, $t^*$ produces a nondimensional mean motion for the primary system that is equal to unity, and $m^*$ is set equal to the sum of the masses of the primaries. Through nondimensionalization, the mass ratio, $\mu = M_2/(M_1 + M_2)$, emerges as a natural parameter that governs the characteristics of the solution space. Using these definitions, the nondimensionalized state of the spacecraft is written in the rotating frame and relative to the system barycenter as $x = [d^T, u^T]^T$ where $d = [x, y, z]^T$ and $v = [\dot{x}, \dot{y}, \dot{z}]^T$ are the position and velocity vectors, respectively. Then, the equations of motion governing the spacecraft in the CR3BP are written as

$$
\ddot{x} - 2\dot{y} = \frac{\partial U^*}{\partial x} \quad \ddot{y} + 2\dot{x} = \frac{\partial U^*}{\partial y} \quad \ddot{z} = \frac{\partial U^*}{\partial z}
$$

where the pseudo-potential function is $U^* = (1/2)(x^2 + y^2) + (1 - \mu)/d_1 + \mu/d_2$ and $d_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $d_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$ are the distances of the spacecraft from each of the primaries. This dynamical model admits a constant of motion labeled the Jacobi constant and equal to $C_J = 2U^* - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$; this quantity is inversely related to the spacecraft energy. Throughout the phase space, a variety of fundamental solutions exist including equilibrium points, periodic orbits, quasi-periodic orbits, and hyperbolic invariant manifolds.

3 BACKGROUND: REINFORCEMENT LEARNING

State-of-the-art RL methods typically train neural networks, which act as universal function approximators. During training, the goal is to learn both the optimal action, $u_t$, to perform at every state, $s_t$, in the environment and the states in the environment that maximize the value function, which is equivalent to maximizing the expected total reward. The mapping that determines the action to perform at every state in the environment is labeled the policy function and denoted as $\pi(\cdot|\cdot)$. The value function is equal to

$$
V^\pi(s_t) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, u_t, s_{t+1})\right]
$$

where $r_t(s_t, u_t, s_{t+1})$ is the immediate reward for the state-action pair $(s_t, u_t)$ at time step $t$ and $\gamma$ is the discount factor which may weight current and near-future rewards higher than more distant rewards. Specific to each application, the reward function is formulated to encapsulate both the objectives and/or constraints within an environment. To learn both the optimal mapping between states and actions and the value function, neural networks interact with an unknown environment by controlling agents that gather state-action-reward experiences in the environment. The neural networks are initialized with no knowledge of the environment and initially act randomly. Then, the goal is for the neural networks to gradually and autonomously converge on locally optimal behavior.
during training without input from a domain expert by using the state-action-reward experiences to update the parameters of the neural network.\textsuperscript{6} This enables RL algorithms to train neural networks to solve complex optimization problems, construct a policy that maps a high-dimensional solution space to suitable actions, and in some cases recover locally optimal solutions that may be non-intuitive to the domain expert.\textsuperscript{28}

Although there are a variety of structures that may be used within an RL algorithm, actor-critic structures possess strong convergence properties and have demonstrated robust performance in complex, high-dimensional environments.\textsuperscript{29} Actor-critic methods use two neural networks to learn the policy and value functions separately, thereby simplifying the learning process and aiding convergence.\textsuperscript{29} Specifically, the actor neural network learns the policy function while the critic neural network approximates the value function.\textsuperscript{29} Additionally, the actor and critic neural networks are typically constructed using distinct architectures, which has been observed to aid convergence.\textsuperscript{30} In this paper, the hyperbolic tangent activation function is used within both the actor and critic neural networks. The actor neural network is formulated with more hidden layers while the critic neural network is constructed using less hidden layers but more nodes per hidden layer.\textsuperscript{30} While actor-critic methods form the foundation for state-of-the-art RL algorithms, there are a variety of RL algorithms that may be used to train neural networks in an actor-critic structure.

PPO is selected as the RL algorithm to train the neural networks in an actor-critic structure in this paper due to its demonstrated reliability and robust performance in sensitive chaotic environments, including the CR3BP.\textsuperscript{8,10,12,15,30,31} PPO simplifies the objective function used by previous RL methods by enforcing a soft constraint on the update size to the parameters of the neural networks.\textsuperscript{32} This objective function is composed of three distinct elements that influence the updates to the networks: (1) a clipped objective that prevents certain state-action pairs from destabilizing the networks, (2) a value error term that penalizes errors in the value function estimation, and (3) an entropy term that balances the exploitation and exploration trade-off within the environment.\textsuperscript{32} First, the clipped objective is defined as

\begin{equation}
L_t^{CLIP}(\theta_j) = \mathbb{E}_t [\min(R_t(\theta_j)\hat{A}^\pi_t, \text{clip}(R_t(\theta_j), -\epsilon, \epsilon)\hat{A}^\pi_t)]
\end{equation}

where

\begin{equation}
R_t(\theta_j) = \pi_{\theta,j}(u_t|s_t) - \pi_{\theta,j-1}(u_t|s_t)
\end{equation}

is the probability difference between the old and new policies which is encouraged to stay near zero to prevent potentially destabilizing updates to the networks. In addition, \(\epsilon\) is a clipping parameter governing the maximum threshold on the probability difference, and \(\hat{A}^\pi_t\) denotes the estimated advantage function which evaluates the benefit of each state-action pair.\textsuperscript{32} In this paper, the advantage function is estimated using Generalized Advantage Estimation (GAE) which measures the difference between the estimated and received value for a state-action pair without introducing significant bias or variance.\textsuperscript{7} Using GAE, the estimated advantage function is calculated as

\begin{equation}
\hat{A}^\pi_t(s_t, u_t) = \sum_{\ell=0}^{\infty} (\gamma \lambda)^\ell \partial_{t+\ell}
\end{equation}

where

\begin{equation}
\partial_t = r_t(s_t, u_t, s_{t+1}) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)
\end{equation}
is the temporal difference residual of the estimated value function and \( \lambda \) denotes the GAE factor which balances the bias-variance trade-off within the estimated advantages.\(^7\) Using these components, the objective function used in PPO is defined as

\[
L^{CLIP+VF+S}_{t}(\theta_j) = \mathbb{E}_t[L^{CLIP}_{t}(\theta_j) - c_1 L^{VF}_{t}(\theta_j) + c_2 S[\pi_{\theta,j}](s_t)]
\]  

(7)

where \( L^{VF}_{t}(\theta_j) = (V_{est}^{\theta}(s_t) - V_{act}^{\theta}(s_t))^2 \) measures the error in the estimated value function, \( S[\pi_{\theta,j}](s_t) \) encourages exploration within the environment, and the scalar coefficients \( c_1 \) and \( c_2 \) weight the value error and entropy terms.\(^32\) Once the objective function is evaluated for a set of state-action-reward experiences, an optimizer updates the parameters of the neural network until convergence onto locally optimal behavior; the implementation in this paper uses AdamW, a variant of the well-known Adam optimizer that decouples selecting the weight decay factor from the update steps.\(^33,34\)

4 APPROACH: RECONFIGURATION AS A REINFORCEMENT LEARNING PROBLEM

RL is used in this paper to design reconfiguration maneuvers that adjust the position of a starshade relative to a space telescope that is located along an \( L_2 \) halo orbit in the Sun-Earth CR3BP. This section begins with an overview of the general scenario used in this paper. Then, the scenario and maneuvering goals are used to formulate an RL problem, requiring definition of the environment, state vector, action, reward function, and episode.

4.1 Scenario Overview

In the scenario explored within this paper, a starshade with maneuvering capability is operating in the vicinity of a space telescope that is assumed to follow an \( L_2 \) southern halo orbit in the Sun-Earth CR3BP. This reference orbit is selected to resemble the mission orbit currently employed in development of the Nancy Grace Roman Space Telescope.\(^17\) Specifically, the reference \( L_2 \) southern halo orbit that locates the space telescope is assumed to possess a period of \( T_{ref} = 180 \) days and a Jacobi constant of \( C_J = 3.00078.\)\(^15\) This reference trajectory, which exists in the Sun-Earth CR3BP, is plotted in black in Figure 1 in the Sun-Earth rotating frame using dimensional coordinates; the \( L_2 \) equilibrium point is located by a gray diamond. Throughout the scenario, the space telescope is assumed to exactly follow the periodic orbit in the CR3BP with no uncertainty or perturbations.

![Reference L2 southern halo orbit with a 180 day period used to locate the space telescope in the Sun-Earth CR3BP.](image)

Figure 1. Reference \( L_2 \) southern halo orbit with a 180 day period used to locate the space telescope in the Sun-Earth CR3BP.
In the actual application motivating this scenario, a starshade must be positioned to block starlight from a specified host star while the instruments onboard the telescope observe nearby exoplanets.\textsuperscript{16} To achieve this goal, the starshade must possess a specific relative position vector that produces a desired separation between the starshade and the space telescope as well as a direction that depends on the target of the observation.\textsuperscript{16,17} The relative position vector is calculated from a required separation that is derived from hardware and mission parameters along with the direction to the host star, specified in an inertial frame. This relative position vector must be maintained, to within allowable tolerances on the drift, for a desired time interval, labeled the observation phase; during this time, no maneuvers are applied. Next, the observing targets change frequently throughout the mission, requiring the starshade to maneuver between distinct relative position vectors between the end of one observation interval and the start of the next observation interval; this intermediate phase is labeled the transfer phase. During the transfer phase, maneuvers are used to target the desired relative position vector at the beginning of the upcoming observation arc. At the end of the transfer arc and immediately before the observation arc, a single maneuver is used to target the desired relative position vector at the end of the observation arc.

To support translating the complex scenario of a starshade frequently applying reconfiguration maneuvers into a simplified RL problem for preliminary analysis, several assumptions are introduced. First, the space telescope is assumed to exactly follow its reference orbit in the natural Sun-Earth CR3BP, i.e., no maneuvers are applied and no deviation from the reference occurs. Next, following the definitions and simplifying assumptions used by several authors studying reconfiguration maneuvers for the starshade, observation of a specific target requires computation of two arcs for the starshade: a transfer arc and an observation arc. The spacecraft is assumed in this paper to apply a sequence of four impulsive maneuvers distributed equally in time across the transfer arc; this transfer arc is assumed to possess a duration between 6 and 21 days.\textsuperscript{16,17} Of course, an alternative number of impulsive maneuvers or continuous maneuvers could be employed in subsequent analyses. One goal in calculating the four impulsive maneuvers assumed in this scenario is to minimize the deviation of the spacecraft relative position from the desired relative position at both the beginning and end of the observation arc; this observation arc is assumed to possess a duration between 0.25 and 5 days.\textsuperscript{16,17} Of course, this assumption applies that the specific relative position vector does not need to be maintained across the entire interval and neglects the relative velocity between the starshade and telescope. An additional significant assumption is introduced in this paper: the desired direction to the host star is assumed to be fixed in the rotating frame during the observation phase. This assumption, used to reduce the complexity of specifying the desired relative position vector over the observation phase, is considered reasonable for this preliminary implementation because the observation phase is constrained to possess a much smaller duration than the period of the primaries relative to the inertial frame. Finally, hard constraints on the deviation from the desired relative position vector are not currently enforced. Despite these assumptions, this simplified relative positioning goal supports the preliminary analysis that is the focus of this paper. The second goal is to reduce the required maneuver magnitude across each transfer arc with the motivation of reducing the total required propellant mass or enabling additional observations.

4.2 Translating Reconfiguration Maneuver Design into an RL Problem

The transfer and observation arcs are used to define an episode. A single episode is assumed in this paper to generally consist of $n$ subarcs or, equivalently, time steps. As depicted conceptually in Figure 2, $n - 1$ blue subarcs are associated with the transfer phase and the final red subarc corresponds to the observation phase. During the transfer phase, each of the $n - 1$ subarcs possesses
an integration time $t_{int,t}$ and begins with a single impulsive maneuver, $\Delta v_i$ for $i = [1, n - 1]$. The final subarc that constitutes the observation phase begins with a single impulsive maneuver, $\Delta v_n$, and possesses a duration of $t_{int,o}$. The relative positioning goal during this observation period is split into two components: minimizing the deviation from both the desired separation between the starshade and space telescope, labeled $d_{sep,des}$, and the desired unit vector $\hat{d}_{des}$.

At each time step, the state vector is defined to capture the information available to the policy and agent. In this formulation, the goal is to train a policy that produces a sequence of maneuvers that guides the starshade to desired relative position vectors for various transfer and observation time intervals, initial conditions, and reconfiguration goals. Accordingly, the agent state vector $s_i$ at the $i$-th time step is defined in this paper as

$$s_i = [\tilde{x}_{ref,i}, \Delta \tilde{x}_i, \tilde{t}_i, \Delta \tilde{t}_i, \hat{d}_{des}, \tilde{d}_{sep,des}] \in \mathbb{R}^{18}$$

where $\tilde{x}_{ref,i}$ is the normalized instantaneous state of the telescope along the reference orbit in the rotating frame, $\Delta \tilde{x}_i$ is the normalized relative state of the starshade with respect to the telescope in the rotating frame, $\tilde{t}_i$ is a normalized integer for the remaining time steps in the episode, $\Delta \tilde{t}_i$ is the normalized integration time along the subarc equal to $t_{int,t}$ for $i = [1, n - 1]$ or $t_{int,o}$ for $i = n$, and $\hat{d}_{des}$ and $\tilde{d}_{sep,des}$ are the desired $3 \times 1$ position unit vector in the rotating frame and scalar separations, respectively, between the starshade and telescope.

During training, each episode is initialized to ensure that the initial deviations and integration times along each arc remain within the specified bounds and that the policy sufficiently covers a suitable array of desired relative position vectors. First, the initial state of the space telescope is randomly selected from the reference Sun-Earth $L_2$ halo orbit. The initial position of the starshade relative to the space telescope is defined by randomly selecting each state component: the order of magnitude is selected using a triangular distribution between 0 and 5 while the scaling factor is selected using a uniform distribution to produce a maximum value of 50,000 km and 10 m/s in each of the position and velocity components, respectively. The integration time along each of the transfer and observation subarcs, i.e., $t_{int,t}$ and $t_{int,o}$, respectively, are also randomly seeded from a uniform distribution.
distribution for each new episode. The unit vector that governs the desired relative configuration of the starshade is calculated from two angles that are randomly selected from a uniform distribution from \([-\pi, \pi]\) rad and \([-\pi/2, \pi/2]\) rad. Finally, the desired separation between the starshade and telescope is selected from a uniform distribution between 30,000km and 45,000km.\textsuperscript{17} The upper and lower bounds on these quantities used during initialization appear in the second and third columns of Table 1. With this initialization strategy, a policy is trained to recover the maneuver sequences required for a variety of reconfiguration transfers. Within the definition for the agent state vector and at each time step, each quantity is normalized to predominantly lie inside the range \([-1, 1]\) to aid the neural network convergence. Aside from the components associated with the desired unit vector, the \(j\)-th normalized component of the observation vector at the \(i\)-th time step, \(o_{i,j}\) is calculated using the following expression:

\[
o_{i,j} = \frac{2(O_{i,j} - O_{j,min})}{O_{j,max} - O_{j,min}} - 1
\]

where \(O_{i,j}\) is the original quantity prior to normalization and \(O_{j,max}\) and \(O_{j,min}\) are the specified upper and lower bounds, listed in the last two columns of Table 1.

The action vector is defined to reflect the impulsive maneuvers that the spacecraft may apply at the beginning of each time step. In the context of the simplified scenario explored within this paper, these actions are defined using a vector with three components \(u_{x,i}, u_{y,i}, u_{z,i}\) in the Sun-Earth rotating frame such that the action vector is equal to

\[
\mathbf{u}_i = [u_{x,i}, u_{y,i}, u_{z,i}] \in \mathbb{R}^3
\]

To calculate the impulsive maneuver \(\Delta v_i\) in the rotating frame from \(\mathbf{u}_i\), a fixed scaling factor \(u_{\text{scale}}\) is specified; in this paper, \(u_{\text{scale}} = 50\) m/s. Although the majority of individual maneuvers tend to

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
Quantity & Initializing Time Steps & Normalization & \\
\hline
\(x_{\text{ref}}\) & Minimum | Maximum & Minimum \((O_{j,min})\) & Maximum \((O_{j,max})\) & \\
\hline
\(y_{\text{ref}}\) & Selected along reference halo orbit & 0.99 & 1.03 & \\
\(z_{\text{ref}}\) & & \(-1 \times 10^{-2}\) & \(1 \times 10^{-2}\) & \\
\(\dot{x}_{\text{ref}}\) & & \(-5 \times 10^{-3}\) & \(4 \times 10^{-3}\) & \\
\(\dot{y}_{\text{ref}}\) & & \(-7.2 \times 10^{-3}\) & \(7.2 \times 10^{-3}\) & \\
\(\dot{z}_{\text{ref}}\) & & \(-2 \times 10^{-2}\) & \(2.2 \times 10^{-2}\) & \\
\hline
\(\Delta x, \Delta y, \Delta z\) & -50,000 km | 50,000 km & -100,000 km | 100,000 km & \\
\hline
\(\Delta \dot{x}, \Delta \dot{y}, \Delta \dot{z}\) & -10 m/s | 10 m/s & -200 m/s | 200 m/s & \\
\hline
\(t\) & \(n\) | \(n\) & \(0\) | \(n\) & \\
\hline
\(\Delta t_{\text{int,t}}\) & 2 | 7 & 0.25 days | 7 days & \\
\hline
\(\Delta t_{\text{int,o}}\) & 0.25 | 5 & 0.25 days | 7 days & \\
\hline
\(d_{\text{sep,des}}\) & 30,000 km | 45,000 km & 30,000 km | 45,000 km & \\
\hline
\end{tabular}
\caption{Upper and lower bounds on the components of the observation vector using in initializing each episode and normalization. Quantities are nondimensional unless otherwise specified.}
\end{table}
possess a magnitude that is below 50 m/s, the action vector components are not constrained between $[-1, 1]$; as a result, individual components may possess larger magnitudes as dictated by the policy.

The reward function is formulated to reflect a simplified set of goals for the starshade transferring from an initial position relative to the space telescope to an observation arc with a specified separation and relative position vector at its beginning and end. Without implementation of more complex RL algorithms that enforce equality and inequality constraints, the reward function is formulated to simultaneously balance three goals: 1) minimizing the total maneuver magnitude across the transfer phase; 2) minimizing the angle between the actual relative position vector and the desired position vector at the beginning and end of the observation phase, labeled $\theta$ and expressed in radians; and 3) minimizing the difference between the actual and desired separation at the beginning and end of the observation phase, labeled $\Delta d$ and expressed in km. Accordingly, the reward function selected for this simplified scenario is defined as

$$r = \begin{cases} 
    (1 - \|u_i\|) & \text{if } i < (n - 1) \\
    (1 - \|u_i\|) - \ln (\theta_i + 1) - \ln (|\Delta d_i| + \epsilon_d) & \text{if } i \geq (n - 1)
\end{cases}$$

(11)

to incentivize distinct behavior during the transfer and objective phases. In this expression, $\epsilon_d = 1 \times 10^8$ is used to ensure that the natural logarithm does not become undefined if $|\Delta d_i| = 0$. Throughout the transfer phase, the policy is rewarded for minimizing the required control effort while targeting $\hat{d}_{des}$. Then, in the observation phase, the reward function balances minimizing the required control effort and minimizing the angle from the desired relative position unit vector and the deviation from the desired separation. Note that the $|\Delta d_i|$ and $\theta_i$ values appear within the natural logarithm function to significantly increase the reward as the values approach zero. Of course, formulating the reward function in this manner does not guarantee satisfaction of any specific hard constraints on the maximum or minimum thresholds on each of these goals. However, the definition used in this paper incorporates each of the goals of the simplified reconfiguration scenario at similar orders of magnitude.

5 RESULTS: RECONFIGURATION MANEUVER DESIGN IN THE CR3BP

The RL problem formulated in Section 4 is used to train a policy that autonomously generates maneuvers for a starshade operating near a space telescope located along a Sun-Earth $L_2$ southern halo orbit in the CR3BP. In this demonstration, the starshade is assumed to perform four impulsive maneuvers per episode. Accordingly, the transfer phase of each episode is composed of three time steps whereas the observation phase consists of one time step. In addition, the dynamical environment during both training and evaluation is modeled ideally with no perturbations or uncertainty incorporated into this preliminary work. This section begins by presenting an overview of the training process, designed to balance learning within a manageable computational time with identifying a policy that achieves the desired relative positioning and maneuvering goals to within a reasonable error. Then, the policy is used to produce the maneuvers required for the starshade to reconfigure for various transfer and observation time intervals as well as various initial and desired relative position vector combinations. Evaluating the policy for a variety of randomly generated and unrelated maneuver sequences supports an initial assessment of the results produced by the RL-based maneuver planner across the broader solution space. Finally, the policy is used to generate the reconfiguration maneuver sequences that successively target a randomly generated list of desired relative position vectors over fixed transfer and observation time intervals; a scenario that reflects a starshade reconfiguring to block starlight from a sequence of observing targets over approximately 200 days.
5.1 Training Process

PPO is governed by parameters that describe the structure of the actor and critic neural networks, the objective function used by PPO, and the updates to the neural network. These governing parameters are often selected using a manual tuning process, grid search, random search, Bayesian optimization, or other optimization schemes such as RL. The values and neural network configurations used to govern the implementation in this paper are selected through manual tuning from a reference combination used in our previous implementation of PPO for maneuver design in the CR3BP; this baseline combination was identified via Bayesian optimization and used to train a policy in a multiple-maneuver station-keeping scenario. Following manual tuning, the selected hyperparameters and neural network structures are listed in Table 2. In this table, the learning rate and PPO clipping parameter are listed as ranges to reflect that these quantities are selected to vary in a step-like manner across updates during training. This variation, described in more detail in the next paragraph, supports balancing a model learning faster with a larger learning rate with focusing on reducing exploration to learn an optimal solution with a lower learning rate. Note that the policies generated by RL algorithms may depend significantly on the hyperparameter selection. As such, alternative hyperparameter and neural network combinations may produce better or worse results. Even with these hyperparameter selections, separate training runs may produce slightly different results due to the stochastic nature of sampling the agent state space during training and the neural network initializations.

Table 2. Hyperparameter and neural network structure values for PPO implementation of the reconfiguration maneuver design problem.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental Steps, $\tau$</td>
<td>4096</td>
</tr>
<tr>
<td>Epochs, $E$</td>
<td>4</td>
</tr>
<tr>
<td>Mini Batches, $M$</td>
<td>6</td>
</tr>
<tr>
<td>Discount Factor, $\gamma$</td>
<td>0.99</td>
</tr>
<tr>
<td>GAE Factor, $\lambda$</td>
<td>0.9</td>
</tr>
<tr>
<td>Clipping Parameter, $\varepsilon$</td>
<td>$[5 \times 10^{-3}, 5 \times 10^{-2}]$</td>
</tr>
<tr>
<td>Value Function Coefficient, $c_1$</td>
<td>0.5</td>
</tr>
<tr>
<td>Entropy Coefficient, $c_2$</td>
<td>$1 \times 10^{-3}$</td>
</tr>
<tr>
<td>Learning Rate, $l_r$</td>
<td>$[1 \times 10^{-4}, 1 \times 10^{-3}]$</td>
</tr>
<tr>
<td>Actor Hidden Layers</td>
<td>2</td>
</tr>
<tr>
<td>Actor Nodes per Hidden Layer</td>
<td>64</td>
</tr>
<tr>
<td>Critic Hidden Layers</td>
<td>1</td>
</tr>
<tr>
<td>Critic Nodes per Hidden Layer</td>
<td>512</td>
</tr>
<tr>
<td>Activation Function</td>
<td>tanh(·)</td>
</tr>
</tbody>
</table>
In this paper, the training process is split into a sequence of smaller training segments that are governed by values of the learning rate and clipping parameter that are successively decreased. This approach is similar to scheduling the values to vary across updates using a step function; selecting a suitable scheduling scheme for use in the constructed implementation is a subject of ongoing work. Nevertheless, for the sequence-based approach to training used in this paper, each training segment spans $1.8 \times 10^7$ steps within the environment. The first training segment uses a learning rate of $1 \times 10^{-3}$ and clipping parameter of $5 \times 10^{-2}$ to coarsely identify a policy that reduces the average discounted cumulative reward significantly, from approximately -25 to -9.5. Then, subsequent training segments are governed by lower learning rates and clipping parameters to reduce exploration and focus on identifying an optimal solution. Without this modification, simply extending the first training segment with these hyperparameters causes chattering in the average discounted cumulative reward. However, using a sequence of 5 training segments with progressively smaller values of the learning rate and clipping parameter, within the bounds listed in Table 2, results in convergence to a policy with an average discounted cumulative reward of -6.6, evaluated across 10,000 randomly initialized episodes.

5.2 Examining Results Across the Broader Solution Space

The trained policy is evaluated using a variety of initial and desired relative position vectors and time intervals for each of the transfer and observation phases to support a broader analysis of both the results generated by PPO and the maneuver design problem. Specifically, randomly generated state vectors are used to initialize each episode with each component of the state vector constrained to lie within the bounds used during training as reported in Table 1. Following initialization, each episode possesses a fixed integration time for the subarcs along the transfer and observation phases as well as the desired separation and relative unit vector. Accordingly, the maneuvers generated for each episode are not expected to globally minimize the total maneuver magnitude to reconfigure to the desired relative position at the beginning and end of the observation arc. These maneuvers are also not designed to exactly achieve the desired relative position vector to within any required tolerance. Rather, the maneuvers are generated for a fixed transfer and observation time with the goal of simultaneously minimizing the maneuver magnitude and deviation from the relative position vector, expressed using angular and range components.

The trained policy is evaluated across 10,000 episodes initialized using randomly-generated 18-dimensional state vectors. Figure 3 displays a) the initial states along the transfer arcs (blue markers) and observation phases (red markers) and b) the integration times along each arc in the transfer and observation phases across these 10,000 episodes. Across this evaluation set, the top of Fig. 4 displays a histogram of the deviation in the actual separation between the starshade and space telescope at the beginning (blue) and end (red) of the observation arc from the desired separation. The horizontal axis of this histogram is represented on a logarithmic scale for visual clarity with 10 equally-distributed bins across each order of magnitude. Figure 4 displays 99.9% of the data, with 11 and 2 episodes producing a deviation in the separation at $t_n$ and $t_{n+1}$, respectively, that is greater than 10,000 km. Meanwhile, the bottom of Fig. 4 displays the angular separation of the starshade relative position vector from the desired position vector at the beginning (blue) and end (red) of the observation arc. For visual clarity, the bins possess a width of 0.1 degrees, depicting 97.7% and 98.8% of the data that produces an angular deviation less than 2 degrees at the beginning and end, respectively, of the observation phase. Finally, Fig. 5 displays the sum of the magnitudes of the three maneuvers that occur along the transfer arc from $t_1$ to $t_3$, equal to $\Delta v_{tot,t_3}$ in blue and the magnitude of the final maneuver $\Delta v_o$ that occurs at the beginning of the observation arc at $t_4$ in red.
Figure 3. Evaluation data with 10,000 randomly initialized episodes: a) initial states for the transfer (blue) and observation (red) phases and b) duration of each phase.

for all 10,000 episodes. The width of the bins in both of these figures is 5 m/s.

Figure 4. Histograms of deviation from desired separation and angle from desired unit vector at the beginning (blue) and end (red) of the observation arc.

Figure 5. Histogram of maneuver magnitudes: $\Delta v_{\text{tot},t}$ capturing the sum of the three maneuvers that occur along the transfer arc in blue, and the magnitude of the final maneuver $\Delta v_o$ that occurs immediately before the observation arc in red.
To assess the capability for the RL-based maneuver planner constructed in this paper to achieve the desired relative separation between the starshade and telescope, the results in Fig. 4 are first examined. Specifically, 94.8% of the generated maneuver sequences place the starshade within 1,000 km of the desired separation at the beginning of the observation phase, randomly selected between 30,000 km and 45,000 km for each episode; for 76.9% of the episodes, this deviation in the separation is less than 200 km. The desired separation at the end of the observation phase follows a similar trend with 82.3% of the maneuver sequences producing a deviation that is less than 200 km at the end of the observation intervals. Although these deviations from the desired separation distance may not fall within strict predefined tolerances, they are sufficiently small to conclude that the trained policy balances reducing the deviation in separation distance at the beginning and end of the observation arc with its other objectives. Further reduction in these separations may be achieved through additional training, alternative hyperparameter selections, or reward shaping. These results also support a preliminary analysis of the solution space and facilitate higher-fidelity analyses using optimization or RL schemes that incentivize and/or enforce constraints.

The second component of examining the results produced by the trained policy focuses on the angular deviation from the desired position vector. First, analysis of the bottom of Fig. 4 reveals that 93.8% of the maneuver sequences place the starshade in a relative position that is within 1 degree of the desired relative position vector at the beginning of the observation arc whereas 39.6% lie within 0.2 degrees. To physically interpret these angular deviations, consider a scenario where the desired separation between the starshade and space telescope is 37,000 km. In that example, a 1 degree angular deviation between the actual and desired relative position vectors corresponds to 645 km while 0.2 degrees corresponds to 129 km. These deviations tend to be on a consistent order of magnitude with the deviations in the separation distance, as displayed in Fig. 4. There is an increase in the number of maneuver sequences that produce a small angular deviation between the actual and desired relative position vectors at the end of the observation arc: 96.9% lie within 1 degree and 48.2% lie within 0.2 degrees. These results in Fig. 4 indicate that the trained policy balances reducing the angular deviations between the actual and desired relative position vectors at the beginning and end of the observation arc.

As an important consideration in maneuver design, the magnitudes of the maneuvers required for each of the transfer and observations phases of reconfiguration are examined using Fig. 5. First, analysis of the blue histogram in Fig. 5 reveals that the majority of the maneuver sequences require between 40 and 150 m/s distributed across the first three impulsive maneuvers. Then, as displayed in the red histogram in Fig. 5, the majority of maneuver sequences require a final maneuver immediately before the beginning of the observation arc that is between 10 and 50 m/s. Although these required total maneuver magnitudes for each phase are significant, recall that the maneuvers are generated for a wide variety of initial and final relative position vectors as well as initial relative velocities, i.e., there is no optimization or analysis to purposefully select the desired relative position vector during the observation.

Evaluating the policy enables analysis of the maneuver magnitudes across the broader solution space. For instance, Fig. 6 displays the total maneuver magnitude for the transfer phase, using the same definition as in Fig. 5a). In this figure, the horizontal axis reflects the duration of the transfer phase, equal to $3\Delta t_{int,t}$, while the vertical axis displays the angular change in the relative position vector from the beginning of the transfer arc, i.e., $t_3$, and the end of the transfer arc, i.e., $t_n$. Then, the total maneuver magnitude across the transfer phase for each maneuver sequence is represented nonuniquely in this two-dimensional space using color. Analysis of this figure reveals that, in
Figure 6. Total transfer maneuver magnitude as a function of the transfer duration and angular separation between the initial relative position vector at $t_1$ and the desired relative position vector at $t_n$.

In general, longer transfer phases on the order of two to three weeks with smaller angular changes in the relative position vector from $t_1$ to $t_n$ tend to result in lower total maneuver requirements during the transfer phase. Alternatively, transfers that require either a larger change in the angle between the relative position vectors across the transfer phase or a shorter duration tend to increase the maneuver requirements. While such results are consistent with expectations, these results demonstrate that the trained policy supports a rapid analysis of the impact of specific variables on parameters of interest. Note, though, that there are a few episodes that require a large total maneuver magnitude during the transfer phase; for instance, the light green marker with a low change in the angle from the desired relative position vector during a transfer phase with a duration of 14 days. Such markers may indicate the existence of outliers that lie in regions of the 18-dimensional space spanned by the agent state vector that are not sufficiently explored or lie at the boundaries of the training set.

The maneuver magnitudes produced by the RL-based maneuver planner are generally consistent with the results of previous studies that use constrained optimization. For instance, Webster and Folta and Farrés and Webster compute reconfiguration maneuvers for a starshade operating 37,000 km from the Nancy Grace Roman Space Telescope given a specific sequence of observations in an ephemeris model of the Sun-Earth system. The optimized sequence of observations encompasses transfer durations between 4 and 34 days and observation durations between 0.23 and 6.14 days. They calculate a single impulsive maneuver at the beginning of the transfer phase between 2.87 m/s and 31.72 m/s and a single impulsive maneuver immediately before the beginning of the observation phase between 2.27 m/s and 39.29 m/s; these maneuvers are calculated by solving a constrained optimization problem with target star directions specified in an inertial frame. Kolemen and Kasdin solve a constrained optimization problem for reconfiguration assuming continuous-thrust in the CR3BP. In their tradespace analysis, they vary the target star locations for a fixed desired separation of 50,000 km and transfer duration of 2 weeks. Their results indicate that the equivalent total maneuver magnitude across this tradespace tends to fall below 150 m/s for the majority of reconfiguration cases. The maneuver magnitudes calculated by evaluating the trained policy for a wide variety of reconfiguration goals and transfer durations in this paper tend to fall within a similar range of values despite the differences in the types and number of maneuvers used. Thus, although the RL-based maneuver planner presented in this paper is not formulated to only minimize the total maneuver magnitude while precisely achieving the relative positioning requirements, the trained
policy successfully encodes a useful mapping from a wide variety of initial and final relative position vectors and transfer and observation phase time intervals to maneuver sequences that balance achieving these goals.

5.3 Application to a Simplified Observing Sequence

The trained policy is evaluated to produce maneuver sequences that successively target a list of desired relative position vectors over fixed transfer and observation phase durations. This example reflects a starshade reconfiguring to block starlight from a sequence of observing targets over the span of approximately 200 days. The observing sequence used in this example is defined by randomly generating: 1) two angles that define \( \hat{d}_{\text{des}} \) and are each constrained to vary by less than 10 degrees between successive targets, and 2) the values of \( t_{\text{int},t} \) and \( t_{\text{int},o} \), respectively. The resulting unit vectors are listed in Table 3 along with the associated durations of the subarcs within the transfer and observation phases. Of course, these values do not correspond to a physical observing sequence. Rather, this example sequence admits a variety of transfer and observation durations as well as changes in the relative position vector that ensure the evaluation set remains within the bounds of the training set. As a result, this example supports demonstration and examination of using the RL-based maneuver planner to produce successive maneuver sequences over a longer time interval. To evaluate the trained policy, the unit vectors and integration times are as follows:

<table>
<thead>
<tr>
<th>Target Number</th>
<th>( \hat{d}_{\text{des}} )</th>
<th>( t_{\text{int},t} ) (days)</th>
<th>( t_{\text{int},o} ) (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([0.6046, 0.4216, 0.6759]^T)</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>2</td>
<td>([0.5642, 0.3106, 0.7650]^T)</td>
<td>4.9401</td>
<td>4.2000</td>
</tr>
<tr>
<td>3</td>
<td>([0.6002, 0.3692, 0.7096]^T)</td>
<td>2.3705</td>
<td>4.9617</td>
</tr>
<tr>
<td>4</td>
<td>([0.6119, 0.3576, 0.7055]^T)</td>
<td>3.7815</td>
<td>3.1000</td>
</tr>
<tr>
<td>5</td>
<td>([0.4868, 0.3468, 0.8017]^T)</td>
<td>4.7491</td>
<td>1.0835</td>
</tr>
<tr>
<td>6</td>
<td>([0.4142, 0.2554, 0.8736]^T)</td>
<td>2.6620</td>
<td>1.6534</td>
</tr>
<tr>
<td>7</td>
<td>([0.4848, 0.3173, 0.8151]^T)</td>
<td>3.1980</td>
<td>0.4766</td>
</tr>
<tr>
<td>8</td>
<td>([0.5879, 0.3162, 0.7446]^T)</td>
<td>2.2194</td>
<td>2.9226</td>
</tr>
<tr>
<td>9</td>
<td>([0.5754, 0.4142, 0.7053]^T)</td>
<td>2.4497</td>
<td>2.4994</td>
</tr>
<tr>
<td>10</td>
<td>([0.6442, 0.4443, 0.6226]^T)</td>
<td>3.1501</td>
<td>2.9929</td>
</tr>
<tr>
<td>11</td>
<td>([0.5918, 0.3310, 0.7350]^T)</td>
<td>5.0124</td>
<td>0.8231</td>
</tr>
<tr>
<td>12</td>
<td>([0.5871, 0.4486, 0.6738]^T)</td>
<td>4.7928</td>
<td>0.3176</td>
</tr>
<tr>
<td>13</td>
<td>([0.5899, 0.3511, 0.7271]^T)</td>
<td>4.0008</td>
<td>1.0477</td>
</tr>
<tr>
<td>14</td>
<td>([0.6083, 0.3495, 0.7126]^T)</td>
<td>5.0253</td>
<td>4.7578</td>
</tr>
<tr>
<td>15</td>
<td>([0.4888, 0.3637, 0.7930]^T)</td>
<td>6.6273</td>
<td>3.1358</td>
</tr>
</tbody>
</table>
vector in row 1 of Table 3 is multiplied by 37,000 km to produce the relative position vector at the initial epoch while the relative velocity is set equal to the zero vector. Then, the state vector \( \mathbf{x} = [1.00805458, 0.00298248, 0.00150418, 0.00102682, 0.00878914, -0.00322358]^T \) is randomly selected along the Sun-Earth \( L_2 \) southern halo orbit to locate the space telescope at the initial epoch. From this initial condition, the trained policy is evaluated to determine the maneuver sequence required to target a relative position vector with a magnitude of 37,000 km and aligned with the unit vector listed in row 2 in Table 3 using the values for \( t_{int,t} \) and \( t_{int,o} \) in that row. Once the trained policy is evaluated, the relative state vector for the starshade and location of the telescope along the reference at the end of the observing phase supplies the initial condition for the beginning of the next transfer phase. This procedure continues for a total of 14 sequential episodes.

Once the trained policy is evaluated using the list of target unit vectors as well as transfer and observation arc durations in Table 3, the properties of the resulting maneuver sequences and path are examined. Specifically, the path of the starshade that is produced by applying the maneuvers successively generated by the RL-based reconfiguration maneuver planner is plotted in Figure 7 in the Sun-Earth rotating frame; the location of the starshade at the beginning of the first target phase is labeled. Then, blue subarcs occur during the transfer phase while red arcs display the observation phase. Black markers locate the beginning of each sequence for clarity. Figure 8 displays: the deviation from the desired separation, \( \Delta d \); angular deviation from the desired relative unit vector, \( \theta \); and the required maneuvers across the transfer phase, \( \Delta v_{tot,t} \), and immediately before the observation phase, \( \Delta v_o \). The deviation in the separation and angle from the desired unit vector are displayed at both the beginning and end of the observation phase, i.e., at \( t_n \) and \( t_{n+1} \), respectively. This figure reveals that the trained policy produces maneuver sequences that successively place the starshade within 0.5 degrees of the desired direction for each relative position vector and approximately 600 km of the desired separation at the beginning and end of each observation phase. Across the 14 successive maneuver sequences, the total maneuver requirements along the transfer phases and immediately before each observation phase remain below 56.4 m/s and 20.8 m/s, respectively. Furthermore, these properties do not collectively drift over successive maneuver sequences. In fact, while the state vectors remain within the bounds of the training set during this evaluation example, the trained policy is able to generate successive reconfiguration maneuver sequences that achieve the goals specified in Section 4.

6 CONCLUSIONS

Reinforcement learning (RL) is used to design reconfiguration maneuver sequences for a starshade near a space telescope located along a Sun-Earth \( L_2 \) southern halo orbit in the CR3BP. In this paper, reconfiguration encompasses two phases: a transfer phase and an observation phase. Then, the goal of a maneuver sequence, distributed across the transfer phase, is to balance: 1) minimizing control effort; and 2) the angular deviation from a specified relative position vector and deviation from the desired separation between the starshade and telescope at the beginning and end of the observation phase. This reconfiguration maneuver design problem is formulated as an RL-problem by defining the agent, action and state formulations, environment, and reward function. This policy then maps the initial states of the starshade and telescope, the duration of transfer and observation phases, and the target relative position vector to the associated maneuver sequence. This policy is trained using Proximal Policy Optimization (PPO) to maximize the total reward for a starshade reconfiguring near a Sun-Earth \( L_2 \) southern halo orbit. The trained policy is evaluated in two scenarios: 1) a large set of randomly-generated and unrelated reconfiguration maneuver sequences that
Figure 7. Path of the starshade reconfiguring throughout the randomly-generated target sequence.

Figure 8. Evaluating the trained policy using a desired separation of 37,000 km and selected target sequence: $\Delta d$ and $\theta$ capturing the deviation from the desired separation and unit vector at the beginning and end of the observation phase, as well as the maneuvers across the transfer phase and immediately before the observation phase.
encompass the high-dimensional space of the state vector and 2) a set of successive reconfiguration maneuver sequences that span approximately 200 days. In both scenarios, the trained policy successfully produces maneuver sequences that achieve the desired relative positioning goals while also minimizing the control effort. Ongoing work is focused on further reducing the separation and angle deviations and the control effort, while also increasing the complexity of the scenario.

7 ACKNOWLEDGMENT

This work was supported by an Early Stage Innovations grant from NASA’s Space Technology Research Grants Program, under NASA grant 80NSSC19K0222. The third author acknowledges support from a NASA Space Technology Research Fellowship.

REFERENCES


