(Preprint) AAS 21-216

DESIGNING IMPULSIVE STATION-KEEPING MANEUVERS NEAR A SUN-EARTH L2 HALO ORBIT VIA REINFORCEMENT LEARNING

Stefano Bonasera[*], Ian Elliott[*], Christopher J. Sullivan[*], Natasha Bosanac[†], Nisar Ahmed[‡], Jay McMahon[†]

Reinforcement learning is used to plan station-keeping maneuvers for a spacecraft operating near a Sun-Earth L_2 halo orbit and subject to perturbations from momentum unloads. This scenario is translated into a reinforcement learning problem that reflects the desired goals, variables and dynamical environment. Proximal Policy Optimization is used to train policies that generate station-keeping maneuvers in the circular restricted three-body problem and a point mass ephemeris model. These policies successfully produce bounded trajectories with small maneuver requirements, motivating further development of autonomous maneuver planning technologies for spacecraft operating in complex gravitational environments.

1 INTRODUCTION

Autonomous maneuver planning in multi-body environments will be a key capability that both enhances and enables future missions. One type of maneuver that may occur throughout the lifetime of a mission is a station-keeping maneuver, which focuses on maintaining bounded motion near a desired path. Existing strategies for planning station-keeping maneuvers for spacecraft operating in a multi-body environment often leverage a combination of dynamical systems theory, optimization, and support from a human trajectory designer.[1–3] These strategies successfully recover efficient maneuvers at the expense of significant human and computational resources. Developing approaches to autonomously plan these station-keeping maneuvers may be invaluable for reducing the operational cost and complexity of operating large observatories, formations, and small satellites.

Reinforcement learning (RL) algorithms have emerged as a tool for autonomously designing a control policy in complex environments. An RL problem is formulated using a policy, producing the action to apply at a certain observation, and an environment that governs the transition from a specific observation-action pair. The observation represents the information the agent perceives from the environment. Another key component of an RL problem is the mathematical definition of the reward function that encodes the immediate reward of selecting an action at a given observation. Using these definitions, the goal is to uncover a policy that maximizes the long-term reward, i.e., the discounted cumulative reward of successive observation-action pairs. RL algorithms have been

[*]Graduate Researcher, Colorado Center for Astrodynamics Research (CCAR), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303

[†]Assistant Professor, Colorado Center for Astrodynamics Research (CCAR), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303

[‡]Assistant Professor, Research and Engineering Center for Unmanned Vehicles (RECUV), Smead Aerospace Engineering Sciences, 3775 Discovery Dr., Boulder, CO 80303

demonstrated to successfully recover policies in a wide variety of applications and environments.[4–6] In astrodynamics, specifically, Das-Stuart et al.,[7] Miller and Linares,[8] Sullivan and Bosanac,[9,10] and LaFarge et al.[11] use state-of-the-art RL implementations to design low-thrust enabled orbit transfers in chaotic environments, while Scorsoglio et al. apply an RL algorithm to the problem of relative motion around periodic orbits.[12] Guzzetti studies the performance of a tabular Monte Carlo implementation of RL for planar orbit maintenance,[13] while Molnar leverages RL in a hybrid approach with dynamical systems theory, to aid in the analysis and design of spatial orbit station-keeping.[14] These successful applications motivate further exploration of the use of RL in astrodynamics activities that currently rely heavily on large optimization problems and a human-in-the-loop.

In this paper, an RL problem is formulated and employed to train policies planning impulsive station-keeping maneuvers for a spacecraft operating near an unstable Sun-Earth L_2 southern halo orbit. In particular, the formulation is motivated by an analysis performed for the Nancy Grace Roman Telescope (formerly the Wide Field Infrared Survey Telescope) by Bosanac et al.[1] This paper first explores the translation of the maneuver design scenario, typically formulated as an optimal control problem, into an RL problem, including the definition of: an appropriate reward function, the agent observations and associated actions, appropriate terminating conditions, and the environment. The resulting RL-based maneuver planner is implemented using Proximal Policy Optimization (PPO) and PyTorch.[15] PPO is employed due to previous demonstrations of favorable convergence properties when computing a policy for a spacecraft in a multi-body gravitational environment.[8–11] Bayesian optimization is employed to guide the selection of suitable hyperparameters and neural network structures that govern the RL-based maneuver planner. Then, the results of this implementation are verified via a comparison to expected results from dynamical systems theory in a simplified scenario where maneuvers are greedily selected in a dynamical environment that is modeled using the circular restricted three-body problem (CR3BP).[1]

The constructed RL-based maneuver planner is demonstrated in two station-keeping scenarios: one with the dynamical environment modeled using the Sun-Earth CR3BP, and the other with a higher-fidelity point mass ephemeris model. In both scenarios, PPO is used to train a policy that produces station-keeping maneuvers that balance long-term minimization of the control effort and boundedness within the vicinity of the reference. The performance of the training process is evaluated in this paper via the effectiveness and confidence of the trained policy over successive updates to reveal successful convergence on a policy that produces a high discounted cumulative reward. Then, the trained policies are applied to a spacecraft that experiences perturbation due to regular momentum unloads applied in random directions. The RL-based maneuver planner successfully designs maneuvers that maintain bounded motion in the vicinity of a Sun-Earth L_2 southern halo orbit: in the first scenario, a total maneuver magnitude of 15.87 m/s is required over a 20 year duration to bound the spacecraft within 390 km and 0.24 m/s of the reference path; and in the second scenario, a total of 5.13 m/s is used over 8.84 years to bound the spacecraft to within 211 km and 0.08 m/s of the reference path. These results offer a foundation for continued exploration of the use of RL in autonomous maneuver planning for spacecraft operating in complex multi-body gravitational environments and, eventually, in more complex maneuver design scenarios.

2 BACKGROUND: DYNAMICAL MODELS

This paper focuses on a spacecraft that is station-keeping near a reference trajectory in the Sun-Earth system using two dynamical models of increasing fidelity: the CR3BP and a point mass ephemeris model. The configuration and equations of motion for each of these models are presented

in this section. In the CR3BP, periodic orbits exist in the rotating frame defined by the two primary bodies. However, in an ephemeris model, these periodic solutions do not exist. Therefore, a periodic orbit that exists in the Sun-Earth CR3BP is used in a multiple-shooting method to recover a nearby, continuous trajectory in an ephemeris model; this procedure is briefly summarized in this section. Each of these paths serve as a reference for the spacecraft during station-keeping maneuver design.

2.1 Circular Restricted Three-Body Problem

The CR3BP describes the motion of an assumed massless spacecraft subject to the gravitational attraction of two primaries modeled as point masses and following circular orbits about their mutual barycenter. The equations of motion for the CR3BP are usually formulated in an orthogonal frame $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ that rotates with the two primaries, P_1 and P_2: the $\hat{\boldsymbol{x}}$-axis is directed from P_1 to P_2, the $\hat{\boldsymbol{z}}$-axis is directed along the orbital angular momentum vector of the system, and the $\hat{\boldsymbol{y}}$-axis completes the right-handed triad. In addition, normalization is often employed via three characteristic quantities: the characteristic length is set equal to the assumed constant distance between P_1 and P_2, the characteristic mass equals the sum of the masses of the primaries, and the characteristic time produces a nondimensional period of the primary system that is equal to 2π. Following normalization, μ represents the mass ratio, equal to $\mu \approx 3.00348064 \times 10^{-6}$ in the Sun-Earth CR3BP. Then, the nondimensional spacecraft state vector relative to the P_1-P_2 barycenter is defined in the rotating frame as $\boldsymbol{x} = [x,\, y,\, z,\, \dot{x},\, \dot{y},\, \dot{z}]^T \in \mathbb{R}^6$. Using these definitions, the nondimensional equations of motion for the spacecraft in the CR3BP and in the rotating frame are written as:

$$\ddot{x} - 2\dot{y} = U_x, \qquad \ddot{y} + 2\dot{x} = U_y, \qquad \ddot{z} = U_z \tag{1}$$

where the pseudo-potential function is defined as $U = (x^2 + y^2)/2 + (1-\mu)/d_1 + \mu/d_2$ and the distances between the spacecraft and the primaries are $d_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $d_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$. When formulated in the rotating frame, the CR3BP is an autonomous dynamical model that admits an energy-like integral of motion, the Jacobi constant, equal to $C_J = 2U - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$. The CR3BP also admits a variety of solutions: five equilibrium points L_i with $i \in \{1, \ldots 5\}$, periodic and quasi-periodic orbits, as well as chaotic motion.[16]

2.2 Point Mass Ephemeris Model

A Sun-Earth point mass ephemeris model represents a higher-fidelity description of the dynamics in the Sun-Earth system. In this paper, this model is formulated to include the gravitational influence of both the Sun and Earth, labeled P_1 and P_2 and each assumed to be spherically symmetric; note that higher fidelity dynamical models may incorporate the gravity due to additional bodies such as the Moon. The mass of the spacecraft is assumed to be negligible in comparison and its gravitational influence on each of the two celestial bodies is ignored. Consistent with the Sun-Earth CR3BP, the same characteristic quantities are used for normalization. In an inertial and orthogonal reference frame $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}})$ and relative to an inertially-fixed basepoint, O, the state of a spacecraft is defined as $\boldsymbol{X} = [X,\, Y,\, Z,\, \dot{X},\, \dot{Y},\, \dot{Z}]^T \in \mathbb{R}^6$. Then, the generic body P_j is located with a nondimensional position vector $\boldsymbol{R}_j = X_j\hat{\boldsymbol{X}} + Y_j\hat{\boldsymbol{Y}} + Z_j\hat{\boldsymbol{Z}}$, while the spacecraft is located via $\boldsymbol{R}_3 = X\hat{\boldsymbol{X}} + Y\hat{\boldsymbol{Y}} + Z\hat{\boldsymbol{Z}}$. Then, the nondimensional position vector of the spacecraft relative to P_j is denoted $\boldsymbol{R}_{j,3} = \boldsymbol{R}_3 - \boldsymbol{R}_j$. In a Sun-Earth point mass ephemeris model, the nondimensional equations of motion for the spacecraft are written in an Earth-centered J2000 inertial coordinate system as:

$$\boldsymbol{R}''_{E,s/c} = -\frac{GM_E}{R^3_{E,s/c}}\boldsymbol{R}_{E,s/c} + GM_S\left(\frac{\boldsymbol{R}_{s/c,S}}{R^3_{s/c,S}} - \frac{\boldsymbol{R}_{E,S}}{R^3_{E,S}}\right) \tag{2}$$

where G is the nondimensional universal gravitational constant, the nondimensional mass of each primary is M_i, and the subscripts S, E, and s/c correspond to the Sun, Earth, and spacecraft, respectively. In this paper, the Jet Propulsion Laboratory DE421 ephemerides are accessed via the SPICE toolkit for both state information and for use in frame transformations.[17,18]

2.3 Recovering a Continuous Trajectory in an Ephemeris Model

Multiple shooting is employed to recover a continuous trajectory that exists in a Sun-Earth point-mass ephemeris model and retains the geometry of a nearby periodic solution from the CR3BP. In this work, the multiple shooting scheme is formulated following the procedure outlined by Bosanac et al.; this approach is briefly summarized here.[1] First, an initial guess is constructed for a desired initial epoch using multiple revolutions of the reference periodic orbit that exists in the CR3BP. The initial guess is formed by discretizing this path into a sequence of arcs: the description of each arc is composed of the initial state that is expressed in an Earth-centered J2000 inertial coordinate system following a coordinate transformation, the associated initial epoch, and the integration time along the arc. This information is combined into a free variable vector. A constraint vector is then formed to reflect state and time continuity between each pair of neighboring arcs. The free variable vector that describes the discontinuous initial guess is then iteratively updated via Newton's method until the constraint vector equals zero to within a small tolerance. This procedure successfully produces a continuous trajectory, associated with a specific initial epoch, that serves as a suitable reference path for a spacecraft operating in the Sun-Earth point mass ephemeris model.

3 BACKGROUND: REINFORCEMENT LEARNING

RL algorithms leverage neural networks, which act as universal function approximators, to learn the optimal policy that maximizes the long-term reward, denoted the value or discounted cumulative reward, returned from an agent interacting in an RL environment.[4] To facilitate efficient and robust learning without extensive support from a human-in-the-loop, the neural networks are structured in an actor-critic setup: the actor neural network learns the optimal actions to take at every observation in the environment and the critic neural network establishes the observations with the highest value in the environment.[19,20] In this paper, the actor and critic neural networks are trained using PPO, an RL algorithm that possesses advantageous convergence properties in chaotic, complex environments.[21–23] This section presents a brief summary of the fundamental components of PPO.

3.1 Feed-forward Neural Networks

Neural networks are universal function approximators for learning complex, nonlinear mappings between inputs and outputs in higher-dimensional environments.[24] A neural network is composed of nodes arranged in consecutive and connected layers enabling outputs from a preceding layer to be fed into the following layer. Feed-forward neural networks, specifically, are constructed using an input layer, one or more hidden layers, and a final layer. In an RL algorithm, the input layer accepts the current observation of the agent and feeds the observation into the first hidden layer of nodes whereby the connections are assigned weights and biases. To incorporate nonlinearity into the network, each of the hidden nodes is assigned a nonlinear activation function, such as the hyperbolic tangent function.[25] Then, the first hidden layer of the neural network is fully connected to a second hidden layer with weights and biases and this process is repeated until the final hidden layer. The final hidden layer is fully connected to an output layer; for an RL algorithm, the product of the output layer may represent the value for the critic neural network, or the characteristics of the distribution

associated with the action sampling for the actor network. The training parameters, comprising the weights and biases for each connection in this feed-forward neural network, are iteratively adjusted using a stochastic gradient descent optimization algorithm based on backpropagation. This approach captures, via the chain rule of derivatives, the influence of each training parameter on the gradient of a defined loss function.[26,27] The implementation in this paper employs the Adam optimization algorithm, which has been demonstrated to efficiently and robustly converge in RL algorithms and complex environments.[21,28]

3.2 Actor-Critic Methods

Actor-critic methods form the foundation of many state-of-the-art RL algorithms by combining the benefits of value-based and policy-based learning methods without significant penalties.[29] In an actor-critic structure, the policy and value function are learned independently by the actor and critic, respectively, to simplify the learning process; one approach involves modeling each of the actor and critic via a feed-forward neural network. The actor neural network approximates the policy function, denoted $\pi_{\boldsymbol{\theta}}(\boldsymbol{u}_t|\boldsymbol{s}_t)$, which maps locally optimal actions, $\boldsymbol{u}_t$, to every observation, $\boldsymbol{s}_t$, in the environment. Concurrently, the critic determines the observations in the environment that are expected to maximize the cumulative reward returned over an episode, which is mathematically encapsulated within the value function. In this work, an episode is composed of a finite number of time steps, each step defined as a single interaction between the actor and the environment.[30] The value function for an observation at time t in the environment is written as:

$$V^{\pi}(\boldsymbol{s}_t) = \sum_{t=0}^{\tau} \gamma^t r_t(\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{s}_{t+1}) \tag{3}$$

where τ represents the maximum number of steps for an episode, γ denotes the discount factor which decreases the influence of future rewards compared to more immediate rewards, and $r_t(\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{s}_{t+1})$ is the reward function for the observation-action pair at time t and the observation at time $t+1$.[31] Although numerous state-of-the-art RL algorithms successfully leverage an actor-critic structure, PPO has been observed to perform well in complex environments.[21,23]

3.3 Proximal Policy Optimization

PPO is a state-of-the-art RL algorithm that has been successfully applied to a variety of multi-body trajectory design scenarios due to its robust convergence in chaotic environments.[8,10,11,21–23] Once a set number of observation-action-reward experiences have been collected from the environment using the neural networks, the experiences are used to form an update to the parameters of the neural networks. First, an advantage function is defined to determine the observation-action pairs in the environment that return the highest value. The advantages of each observation-action pair are estimated using Generalized Advantage Estimation (GAE) via

$$\hat{A}_t^{\pi}(\boldsymbol{s}_t, \boldsymbol{u}_t) = \sum_{\ell=0}^{\infty} (\gamma\lambda)^{\ell} \delta_{t+\ell} \tag{4}$$

where

$$\delta_t = r_t(\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{s}_{t+1}) - V^{\pi}(\boldsymbol{s}_t) + \gamma V^{\pi}(\boldsymbol{s}_{t+1}) \tag{5}$$

is the estimated advantage of the action $\boldsymbol{u}_t$ and λ is the GAE factor, which influences the bias-variance trade-off in the estimated advantages.[31] Then, PPO incorporates a trust region constraint

inspired by another RL algorithm, Trust Region Policy Optimization, to prevent large changes from destabilizing the neural networks.[32] In this paper, the change in the neural networks for an update j is measured using the probability difference between the old and new policies, written as

$$R_{i,t}(\boldsymbol{\theta}_{i,j}) = \pi_{\boldsymbol{\theta},i,j}(\boldsymbol{u}_t|\boldsymbol{s}_t) - \pi_{\boldsymbol{\theta},i,j-1}(\boldsymbol{u}_t|\boldsymbol{s}_t) \tag{6}$$

where values near zero correspond to smaller updates to the neural networks.[9,33,34] Then, a loss function is necessary to compute updates to the parameters of the neural networks using the stochastic gradient descent algorithm and is defined as

$$L_t^{CLIP}(\boldsymbol{\theta}_j) = \hat{\mathbb{E}}_t[\min(R_t(\boldsymbol{\theta}_j)\hat{A}_t, \text{clip}(R_t(\boldsymbol{\theta}_j), -\varepsilon, \varepsilon)\hat{A}_t)] \tag{7}$$

to constrain the probability difference to possess a magnitude less than or equal to the clipping parameter, ε.[22] Two additional terms are appended to the loss function to incentivize continuous exploration in the environment and guide the networks to uncover the value function for the environment.[22] The resulting modified objective function used in PPO is written as

$$L_t^{CLIP+VF+S}(\boldsymbol{\theta}_j) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\boldsymbol{\theta}_j) - c_1 L_t^{VF}(\boldsymbol{\theta}_j) + c_2 S[\pi_{\boldsymbol{\theta},j}](\boldsymbol{s}_t)] \tag{8}$$

where $S[\pi_{\boldsymbol{\theta},j}](\boldsymbol{s}_t)$ is an entropy term that influences the amount of exploration within the environment, $L_t^{VF}(\boldsymbol{\theta}_j) = (V_{\boldsymbol{\theta}}(\boldsymbol{x}_t) - V_t^{target})^2$ is a squared-error loss term that encourages the networks to learn the value function, c_1 is the squared-error loss coefficient, and c_2 is the entropy coefficient.[22] This modified objective function is used by PPO in a stochastic gradient descent optimization algorithm, such as Adam, to train the neural networks using experiences within the environment.[21,22,28]

During training, PPO is primarily governed by a variety of hyperparameters. For the implementation presented in this parameter, hyperparameters of interest include the clipping parameter, the squared-error value loss coefficient, the entropy coefficient, the GAE factor, the discount factor, and the learning rate that governs how quickly the policy is updated by the Adam optimizer. Two additional hyperparameters include the number of epochs specifying how many times each observation-action pair is used to update the parameters and the number of batches defining the amount of groups the observation-action pairs are divided into for each epoch, which both govern how the generated observation-action-reward experiences are accessed for each update.[9] When these hyperparameters are selected appropriately, PPO admits an efficient and robust training process that uncovers locally optimal behavior in a complex dynamical environment.

4 STATION-KEEPING MANEUVER DESIGN VIA REINFORCEMENT LEARNING

The RL-based maneuver planner is demonstrated in the context of a spacecraft station-keeping near a southern L_2 halo orbit in the Sun-Earth system. An overview of this scenario, modeled after the upcoming Nancy Grace Roman Space Telescope, is first presented. An existing station-keeping strategy that leverages dynamical systems theory is also summarized. Then, the procedure for designing station-keeping maneuvers is translated into an RL problem via the definition of the states, actions, reward function, and episode termination criteria. To support the proof of concept presented in this paper, this RL problem is formulated for two approximations of the dynamical environment in the Sun-Earth system: one modeled via the CR3BP and the other using a Sun-Earth point mass ephemeris model. In addition, hyperparameter selection is performed via Bayesian optimization. Finally, a verification of the implementation presented in this paper is presented in a simplified scenario via a comparison to the expected results from dynamical systems theory.

4.1 Scenario Overview

In this paper, the maneuvering spacecraft is assumed to operate near a Sun-Earth L_2 southern halo orbit in a scenario that is modeled after the Nancy Grace Roman Space Telescope. When modeling the dynamics via the CR3BP, a periodic L_2 southern halo orbit with a period of $T_r = 180$ days and a Jacobi constant of $C_J = 3.00078$ is employed as a reference trajectory; this path is displayed in Figure 1(a) in the Sun-Earth rotating frame using dimensional coordinates relative to the Earth. The corrections scheme summarized in Section 2.3 is then used to recover a nearby continuous quasi-halo trajectory that exists in the Sun-Earth point mass ephemeris model. This reference trajectory, associated with an initial epoch of e_0 = 29389.905 MJD and a final epoch e_f = 32628.759 MJD that is approximately 8.87 years later, is displayed in Figure 1(b) in the Sun-Earth rotating frame using dimensional coordinates, centered at the Earth. This reference trajectory geometrically resembles the periodic L_2 southern halo orbit that exists in the CR3BP.

Regular impulsive station-keeping maneuvers are often employed to mitigate the impact of uncertainties, momentum unloads, and off-nominal maneuvers on the spacecraft path. In this paper, regular momentum unloads are the only perturbations considered and are modeled as an instantaneous change in velocity, denoted $\Delta \boldsymbol{v}_{MU}$; full state knowledge and nominal maneuver performance are assumed. Similar to the scenario presented by Bosanac et al. that is based on an early iteration of the Nancy Grace Roman Space Telescope mission parameters, these momentum unloads are assumed to occur in a random direction with a fixed magnitude of 8.7 mm/s; although Bosanac et al. apply these maneuvers every 130 hours, this paper assumes they occur every $t_{MU} = 110$ hours to produce a near-integer ratio with the period of the reference orbit.[1] The station-keeping maneuvers and momentum unloads are assumed to occur at evenly-spaced time intervals, forming momentum unload cycles.[1] Each cycle begins with an impulsive station-keeping maneuver that is designed to maintain boundedness in the vicinity of the reference trajectory; in this paper, there is no constraint on its magnitude and direction. This station-keeping maneuver is followed by a coast arc with a duration of t_{MU}. Three momentum unloads and three coast arcs then occur in alternating order to form a single momentum unload cycle. Following the scenario presented by Bosanac et al., the goal is to design station-keeping maneuvers that require less than 5 m/s per year for up to 10 years.

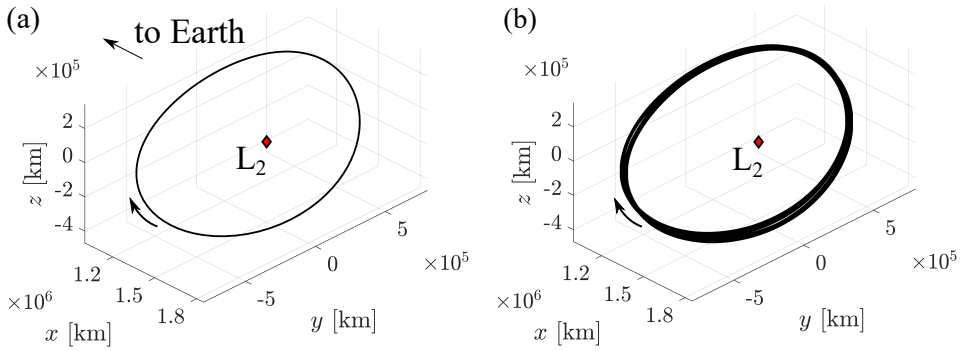

Figure 1. Reference paths near Sun-Earth L_2: (a) southern halo orbit in the CR3BP and (b) nearby quasi-halo trajectory in the Sun-Earth point mass ephemeris model.

4.2 Example of Existing Approach to Designing Station-Keeping Maneuvers

One existing approach to designing low-cost station-keeping maneuvers for a spacecraft operating near a libration point orbit leverages insights from dynamical systems theory. This approach is

based on an observation by Pavlak and Howell in the context of the Acceleration, Reconnection, Turbulence, and Electrodynamics of the Moon's Interaction with the Sun (ARTEMIS) mission. This mission involved two spacecraft following separate reference quasi-halo orbits near each of the Earth-Moon L_1 and L_2 equilibrium points. Pavlak and Howell observed that station-keeping maneuvers that occur at the xz-plane crossings of the rotating frame and are designed to minimize the maneuver magnitude while maintaining long-term boundedness near the reference tend to closely align with the eigenvectors of the monodromy matrix that are associated with the stable mode.[35] More recently, Farrés et al. have further explored this observed alignment between the locally-optimal station-keeping maneuvers and the eigenvector associated with the stable mode.[36]

Bosanac et al. use Pavlak and Howell's observation to reduce the computational complexity of designing station-keeping maneuvers for the Nancy Grace Roman Space Telescope via optimization. At each maneuver location, the greedy station-keeping maneuver design procedure begins by computing the state transition matrix propagated for one period along the perturbed trajectory and calculating the eigenvector associated with the stable mode. Then, the position components of the stable eigenvector are used to define an initial guess for the direction of the impulsive station-keeping maneuver that immediately minimizes the maneuver magnitude while constraining the x-coordinate at the second subsequent xz-plane crossing to within 150 km of the reference. This constrained optimization problem is solved at each maneuver location.[1] Following this approach, Bosanac et al. recover low-cost station-keeping maneuvers in a Sun-Earth-Moon point mass ephemeris model with solar radiation pressure and subject to regular perturbations from momentum unloads; a total maneuver magnitude of 2.2 m/s over a 10 year time interval is used to maintain boundedness near a Sun-Earth L_2 quasi-halo orbit in an ephemeris model. This dynamical systems based approach is used later in this section to verify the RL-based maneuver planner in a simplified control scenario.

4.3 Translating Station-Keeping Maneuver Design into an RL Problem

Three scenarios are formulated in this paper to demonstrate the capability of RL to train a policy for designing station-keeping maneuvers for a spacecraft operating near the selected Sun-Earth L_2 southern halo orbit. The three scenarios employed in this paper are summarized as follows:

- *Scenario 1, greedy station-keeping in the CR3BP*: the spacecraft operates in the CR3BP and the maneuvers are selected individually to greedily minimize deviations from the reference via a constraint on the second subsequent xz-plane crossing.
- *Scenario 2, long-term station-keeping in the CR3BP*: the spacecraft operates in the CR3BP and the maneuvers are selected to ensure the path remains near the reference over a longer episode composed of multiple sequences of maneuvers and coast arcs.
- *Scenario 3, long-term station-keeping in an ephemeris model*: the spacecraft dynamics are governed by the Sun-Earth point mass ephemeris model and the maneuvers are designed to produce a path that remains near the reference over a longer episode, defined using multiple sequences of maneuvers and coast arcs.

These three scenarios reflect increasing levels of fidelity and complexity. The simplest scenario is used solely for verification of the results via a comparison to expected maneuver directions from dynamical systems theory. The second and third scenarios enable demonstration of a more complex, long-term maneuver planning scheme in both low- and higher-fidelity dynamical models.

An RL implementation of the station-keeping maneuver design problem in each of the three scenarios is formulated using a similar foundational structure. During training, a single episode

is defined using up to a maximum number of time steps, τ. Each time step begins by specifying an initial observation-action pair. The continuous observation, $\boldsymbol{o}$, is defined in each scenario to reflect the state of the spacecraft in the dynamical environment and to ensure an efficient mapping to suitable actions. During training, the dynamical environment is modeled without perturbations; momentum unloads are only employed to perturb the path of the spacecraft during evaluation of the trained policies in the long-term station-keeping scenarios. The components of the observation vector, possessing a unique definition in each scenario, are also scaled to generally remain within the range $[-1, 1]$ during training; scaled quantities are indicated using tilde notation throughout this section. At the beginning of the first time step within an episode, the observation is randomly sampled from a continuous uniform distribution across the interval $[-1, 1]$. The continuous action, $\boldsymbol{u}$, is formulated as a 3×1 vector that is scaled to produce the components of the impulsive maneuver, $\Delta\boldsymbol{v}$, in the Sun-Earth rotating frame; in each scenario, this scaling factor is set to $\alpha_A \approx 0.3$ m/s. Then, this observation-action pair is used to form a six-dimensional state vector for the spacecraft at the beginning of the time step. This state vector is then propagated in a selected dynamical environment for a specified duration, Δt, to produce the state of the spacecraft at the end of the time step. The information associated with the path of the spacecraft at the end of this time step is used to evaluate the immediate reward. In the station-keeping maneuver design problem, this reward is formulated as a weighted sum of two objectives: minimizing the control effort and minimizing a specified definition of the deviation from the reference path. When $\tau > 1$, the observation at the end of the current time step seeds the initial observation for the subsequent time step. The propagation and reward evaluation procedure continues until either the specified maximum number of time steps are reached along the episode or additional termination criteria are satisfied. Although the foundational structure of the RL implementation is consistent across the three scenarios, the definitions of the observation vector, reward, and episode are specific to each scenario.

4.3.1 Scenario 1: Greedy Station-Keeping in the CR3BP The first, simplified scenario reflects greedy station-keeping maneuver design in a low-fidelity dynamical environment with boundedness defined using the x-coordinate of xz-plane crossings. This scenario is used solely to facilitate verification of the RL implementation via comparison to expected results from dynamical systems theory. First, the environment used to propagate the spacecraft state during training and evaluation is defined as the natural Sun-Earth CR3BP with no perturbations. Each episode is composed of a single time step corresponding to one station-keeping maneuver, followed by a coast arc. At the beginning of the time step, a relative state vector is denoted as $\delta\tilde{\boldsymbol{x}} = [\delta\tilde{x}, \delta\tilde{y}, \delta\tilde{z}, \delta\dot{\tilde{x}}, \delta\dot{\tilde{y}}, \delta\dot{\tilde{z}}]^T$, defined in the Sun-Earth rotating frame and measured relative to a state along the reference orbit, $\boldsymbol{x}_{ref}$, that is fixed for a single policy. The scaling coefficients used to produce $\delta\tilde{\boldsymbol{x}}$ correspond to dimensional values of $\alpha_P \approx 150$ km and $\alpha_V \approx 0.003$ m/s for position and velocity components, respectively. Accordingly, the observation vector $\boldsymbol{o}_{c,1}$ is defined in this simplified scenario as

$$\boldsymbol{o}_{c,1} = [\delta\tilde{\boldsymbol{x}}] \in \mathbb{R}^6 \tag{9}$$

This observation is used to generate an action from the policy and a relative state in the Sun-Earth rotating frame. The nondimensional relative state is added to the selected initial state vector along the reference orbit. Once an action is applied, the resulting state vector is propagated forward for a time $\Delta t = 1.5T_r$ and the second subsequent crossing of the $xz-$plane is recorded; at this crossing, the displacement in the x-coordinate from the reference orbit is denoted as Δx_I. Then,

the immediate reward is defined to balance minimizing Δx_I and the control effort as

$$r = \begin{cases} -\ln\left(\Delta x_I^2\right) + K\left(1 - \|\boldsymbol{u}\|\right) & \text{if crosses } xz\text{-plane twice within } \Delta t \\ -10 & \text{otherwise} \end{cases} \tag{10}$$

with a scaling coefficient $K = 7$ used in this paper. When the perturbed path of the spacecraft deviates significantly from the reference, the trajectory may not complete two crossings of the xz-plane within the specified integration time. In this case, the observation-action pair that produces this undesirable behavior is penalized by assigning a reward $r = -10$.

4.3.2 Scenario 2: Long-Term Station-Keeping in the CR3BP In this scenario, a station-keeping maneuver is selected as the first maneuver from a sequence of $\tau > 1$ maneuvers, separated by a duration of Δt, that together minimize the cumulative deviation from the reference path and the cumulative control effort. That is, unlike the simplified first scenario, a station-keeping maneuver is not designed to greedily achieve these goals. To translate this scenario into an RL problem, an episode is defined to consist of τ time steps, each composed of an impulsive maneuver followed by a coast arc with a duration, Δt, that is set equal to T_r/τ. Each episode is initialized by randomly selecting a state that lies along the reference halo orbit, denoted $\boldsymbol{x}_{ref}$, and randomly sampling a relative state vector, $\delta\tilde{\boldsymbol{x}}$, from a continuous uniform distribution within the interval $[-1, 1]$. Accordingly, an observation vector is defined in this scenario to reflect both the relative state information and the associated state along the reference orbit as

$$\boldsymbol{o}_{c,2} = [\tilde{\boldsymbol{x}}_{ref}, \delta\tilde{\boldsymbol{x}}] \in \mathbb{R}^{12} \tag{11}$$

Using this observation, the position and velocity components of $\tilde{\boldsymbol{x}}_{ref}$ that lie within the interval $[-1, 1]$ are scaled using the minimum and maximum values of the associated state components along the reference orbit to produce $\boldsymbol{x}_{ref}$. The relative state $\delta\boldsymbol{x}$ is recovered by scaling the last six components of the observation by coefficients that correspond to dimensional values of $\alpha_P \approx 150$ km and $\alpha_V \approx 0.003$ m/s for position and velocity components, respectively. The observation $\boldsymbol{o}_{c,2}$ is input to the policy to generate an action and combined to produce the state vector of the spacecraft at the beginning of a time step. Following propagation for a duration of Δt, the resulting observation-action pair is used to evaluate the immediate reward, defined as

$$r = \begin{cases} -\ln\left(\|\delta\boldsymbol{x}(t_I + t_0)\|\right) + K\left(1 - \|\boldsymbol{u}\|\right) & \text{if } \|\delta\boldsymbol{x}(t_I + t_0)\| \leq 4.5 \times 10^{-5} \\ -100 & \text{otherwise} \end{cases} \tag{12}$$

where $K = 100$ and $\delta\boldsymbol{x}(t_I + t_0)$ is the state deviation of the perturbed path at the end of the time step from the closest state along the reference orbit. To penalize significant departure from the reference orbit, an observation-action pair is assigned a value $r = -100$ if the state deviation at the end of a step exceeds a specified threshold; in this case, the episode is also terminated. The episode continues until either τ steps are performed or a subsequent step generates a reward of $r = -100$.

4.3.3 Scenario 3, Long-Term Station-Keeping in an Ephemeris Model This scenario is formulated consistent with the second scenario, with some modifications to accommodate the use of the Sun-Earth point mass ephemeris model. Each episode is composed of up to τ steps, with each step consisting of a single impulsive maneuver followed by a coast arc with a duration Δt that is equal to T_r/τ. Each episode is initialized by selecting a random initial epoch, t_E, in the range $[e_0, e_f]$ MJD to produce an associated state vector, $\tilde{\boldsymbol{x}}_{ref}$, along the quasi-halo reference that exists in the ephemeris model over the same interval of epochs. A relative state vector, $\delta\tilde{\boldsymbol{x}}$, measured

from the selected reference state and in the rotating frame is then generated by randomly sampling from a continuous uniform distribution across the interval $[-1, 1]$. This information, specified in the Sun-Earth rotating frame, is combined to form the following observation vector:

$$\boldsymbol{o}_e = [\tilde{\boldsymbol{x}}_{ref}, \delta\tilde{\boldsymbol{x}}, \tilde{t}_E] \in \mathbb{R}^{13} \tag{13}$$

This observation vector is leveraged to generate an action from the policy, as well as to compute the initial state of the spacecraft using a scaling approach that is consistent with the second example. The immediate reward for this observation-action pair is then evaluated consistent with the reward function in Scenario 2, defined in Equation (12), and using the same termination conditions. The only difference, however, is that the displacement vector $\delta\boldsymbol{x}(t_I + t_0)$ is computed as the full state deviation between the spacecraft state at the end of the time step and the state along the reference trajectory at the same epoch. After the initial step, the episode continues until either τ steps are performed or a subsequent step generates a reward of $r = -100$.

4.4 Selecting the Hyperparameters Governing the RL Implementation

The performance and results produced by PPO are often significantly influenced by a set of governing hyperparameters and the structure of the underlying neural networks. The hyperparameters are typically selected to achieve an efficient training process that balances both exploration and exploitation to converge on a solution that maximizes the discounted cumulative reward. A variety of strategies for tuning these quantities exist, including: random search, grid search, Bayesian optimization, and leveraging an outer RL-based loop.[37,38] In this paper, Bayesian optimization is employed due to its tendency to exhibit sample efficiency for problems with expensive cost function evaluation, such as those that require numerical integration in sensitive dynamical systems. Specifically, Bayesian optimization describes a generic cost function via a Gaussian process.[37,39] The goal of the optimizer is then identifying a set of inputs that maximize the cost function; in this case, the inputs are selected parameters governing the algorithm. As more input sets are explored, the algorithm increases its confidence of a specific range of inputs to explore in the next iteration. An acquisition function then selects the input set for the next cost function evaluation by balancing the exploration of areas with large uncertainty with the exploitation of regions where the expected mean cost function is large. This process continues until a specified termination condition is satisfied.

This work leverages Bayesian optimization to select a suitable set of hyperparameters and neural network structures that are used in the RL-based maneuver planner. Specifically, PPO hyperparameters and the width and depth of the neural network (NN) structures are selected in the context of the second scenario, focused on long-term station-keeping in the CR3BP. To reduce the required computational effort, this same set of hyperparameters and neural network structures are then employed in all three scenarios. Of course, this single set of governing quantities does not necessarily represent an optimal set in each individual scenario. However, the CR3BP offers a good approximation of the dynamical environment in a higher-fidelity point mass ephemeris model; thus, it is assumed that the quantities derived in one scenario will result in a training process that delivers sufficient results in the other scenarios. Implementation of this Bayesian optimization approach to selecting the hyperparameters and neural network structures leverages the python toolbox *BayesianOptimization*, using the Mattern kernel and the *upper confidence bound* acquisition function.[40] Based on the work by Andrychowicz et al. and Sullivan and Bosanac, the neural networks leverage *tanh* activation functions between consecutive layers and orthogonal initializers across all three scenarios; thus, these properties are not tuned.[9,21] The cost function used to select these parameters during Bayesian

optimization is composed of two terms: 1) the average discounted cumulative reward of the last update batch set and 2) the mean derivative of the average discounted cumulative reward over the last 10 updates, approximated via forward finite differences. This definition favors policies that are both effective in the final update and have exhibited large improvements within at least the final several updates. Using these inputs and cost function definition, Bayesian optimization is applied over a total of 130 runs of the training process: the first 50 training runs correspond to evaluation of random sets of input parameters, while the next 80 runs reflect iterations according to the acquisition function. During optimization, each iteration of the training process is implemented with a variable number of episodes that correspond to a total of 2×10^6 time steps. The optimal hyperparameters and neural networks structures recovered via this procedure are summarized in Table 1. These values are consistent with the general suggestions from authors applying PPO to a variety of complex dynamical environments.[9,21–23,31] Moreover, it appears that in this particular scenario and for the specified cost function used for the optimization, the actor neural network favors multiple layers with a small number of nodes, while the critic neural network benefits from one wide layer.

Table 1. Selected hyperparameters and neural network parameters.

Parameter	Value	Parameter	Value
Learning rate, Lr	5×10^{-3}	Discount factor, γ	0.99
Clipping parameter, ϵ	0.02	GAE factor, λ	0.99
Number of epochs, N_{ep}	4	Actor NN depth, D_{act}	3
Number of batches, N_b	6	Actor NN width, W_{act}	16
Value coefficient, c_1	1×10^{-3}	Critic NN depth, D_{cri}	1
Entropy coefficient, c_2	7×10^{-3}	Critic NN width, W_{cri}	1024

4.5 Verifying the Results of the RL Implementation in a Simplified Scenario in the CR3BP

Expected maneuver directions, derived from dynamical systems theory, are used to verify the solutions produced by the RL implementation in the simplest scenario, focused on greedy station-keeping maneuver design near a Sun-Earth L_2 southern halo orbit in the CR3BP. Four policies are individually trained in this example. Each policy is trained in the CR3BP with the same goal of balancing minimization of the control effort and the deviation in the x-coordinate at the second subsequent crossing of the xz-plane. However, the four policies differ in the definition of the fixed state along the reference used to specify the initial conditions: the reference states are located near each of the extrema in the $z-$ and $y-$directions. Using the hyperparameters listed in Table 1, the training process successfully converges on a solution that possesses a large cumulative discounted reward over 555 updates. Throughout the learning process, the average discounted cumulative reward increases across updates until reaching a plateau; simultaneously, the standard deviations associated with the Gaussian distribution for sampling the actions steadily decrease. This behavior during training is presented and examined in detail in the context of the second more complex scenario.

Following training, each policy is evaluated using a set of initial perturbations from the associated state along the reference orbit to generate individual station-keeping maneuvers that target the second subsequent crossing of the xz-plane while also minimizing control effort. For each trained policy, 30 initial perturbations are selected from a continuous uniform distribution within the range

$[-1, 1]$ and scaled by the α_P and α_V values to produce observation vectors. The resulting maneuvers, calculated from the actions output by each policy, are displayed in Figure 2. In this figure, the reference L_2 southern halo orbit is depicted as a black continuous path in the Sun-Earth rotating frame using dimensional coordinates. Recall that each policy corresponds to a distinct initial reference state along the halo orbit. Thus, 30 arrows represent the maneuvers generated by evaluating each policy near each of the four distinct locations along the reference orbit. For visual clarity, these maneuvers are plotted in a unique shade of purple according to the initial reference state and scaled to depict only the relative magnitude. The zoomed-in views on the right of Figure 2 each display the 30 maneuvers that are associated with a single policy, labeled as A_i, with $i \in \{1, ..., 4\}$. Overlaid on this figure are red arrows that indicate the position components of the stable eigenvector of the monodromy matrix evaluated along the reference periodic orbit. At each location along the orbit, two red arrows appear: both arrows lie in the stable eigenspace but with opposite directions. Analysis of Figure 2 reveals that all four trained policies produce impulsive station-keeping maneuvers that are closely aligned with the position components of the local stable eigenvector. This result is consistent with Pavlak and Howell's observation that station-keeping maneuvers that minimize the maneuver magnitude while targeting a subsequent crossing of the xz-plane tend to align closely with the stable eigenvector of the state transition matrix evaluated along the reference orbit for one revolution. Accordingly, the results presented in Figure 2 indicate that the RL formulation of the station-keeping problem implemented within this paper successfully produces maneuvers in this simplified scenario that are consistent with the expected results from dynamical systems theory.[1,18]

5 RESULTS: STATION-KEEPING MANEUVER DESIGN IN THE CR3BP

RL is used to design station-keeping maneuvers for a spacecraft operating near an L_2 southern halo orbit in the CR3BP. In this scenario, a station-keeping maneuver is designed as the first maneuver from a sequence of 10 maneuvers over 10 time steps that collectively minimize the cumulative deviation from the reference path and control effort. Using the problem formulation outlined in Section 4.3.2 with $\tau = 10$ maneuvers and time steps per episode as well as the hyperparameters in Table 1, PPO is used to train a policy in the natural CR3BP with no perturbations. Training continues for 1×10^7 time steps within this dynamical environment, corresponding to at least 1×10^6

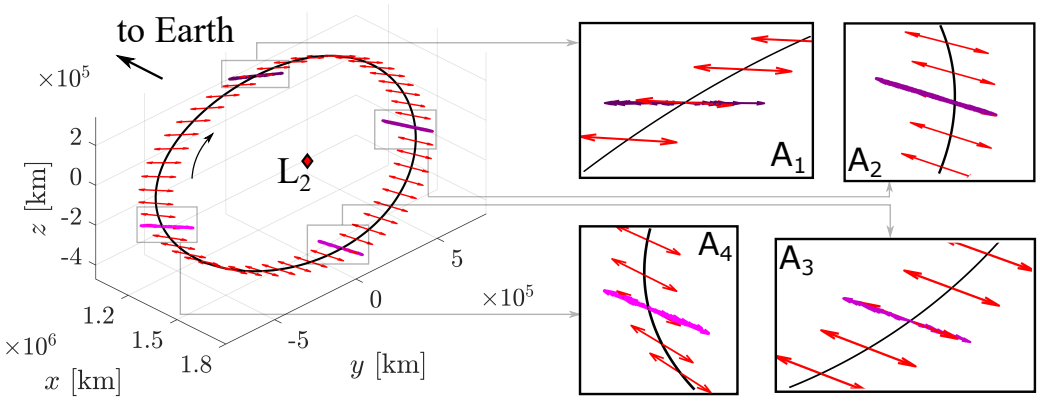

Figure 2. For greedy station-keeping in the CR3BP relative to a Sun-Earth L_2 halo orbit (black), the maneuvers (arrows in shades of purple) generated by the four trained policies tend to align with the stable eigenvectors (red arrows).

episodes. In this particular scenario and for the selected reward function, a straightforward heuristic for expected maneuver directions does not currently exist for verification of the results produced by the RL implementation. Accordingly, further examination of the training process is warranted.

The training process is examined using the discounted cumulative reward and standard deviations of the actions associated with the policy across updates. Figure 3(a) displays the average value of the discounted cumulative reward in black while the shaded red region, labeled $\pm 1\sigma$, corresponds to one standard deviation around the average; the standard deviation is computed using the discounted cumulative rewards in each batch at each update. Analysis of this figure reveals an increase in the discounted cumulative reward and a reduction in the associated standard deviation across updates with a plateau forming towards the end of the training process. This behaviour indicates that the policy is successfully learning to achieve the intended goals encapsulated within the reward function. Throughout the training process, the policy is also improving its understanding of the actions that maximize the discounted cumulative reward. This improvement is visualized via the standard deviations of the Gaussian distribution used to sample the actions, displayed in Figure 3(b). The natural logarithms of the standard deviations for each of the three components of the actions, as defined in the rotating frame, are displayed in distinct colors. The steady decrease of the standard deviations across updates indicates that the policy is gaining confidence in the actions that maximize the discounted cumulative reward by reducing the width of the sampled density functions during training.

The trained policy is evaluated with a variety of initial conditions in the natural Sun-Earth CR3BP to verify that the maneuvers successfully produce controlled trajectories that remain in the vicinity of the reference halo orbit. Specifically, 100 relative state vectors are randomly generated relative to various locations along the reference orbit. The uncontrolled trajectories associated with these initial conditions are propagated in the natural CR3BP for one orbital period and are displayed in Figure 4(a) in the Sun-Earth rotating frame using dimensional coordinates relative to the Earth to demonstrate departure from the reference without any maneuvers; in this figure, Sun-Earth L_2 is located by a red diamond. These same relative state vectors are input to the trained policy to design the first station-keeping maneuver. Following application of the maneuver, the resulting state is propagated in the natural CR3BP and without any perturbations from regular momentum unloads to produce the state at the next time step; the next action is selected by reevaluating the trained policy.

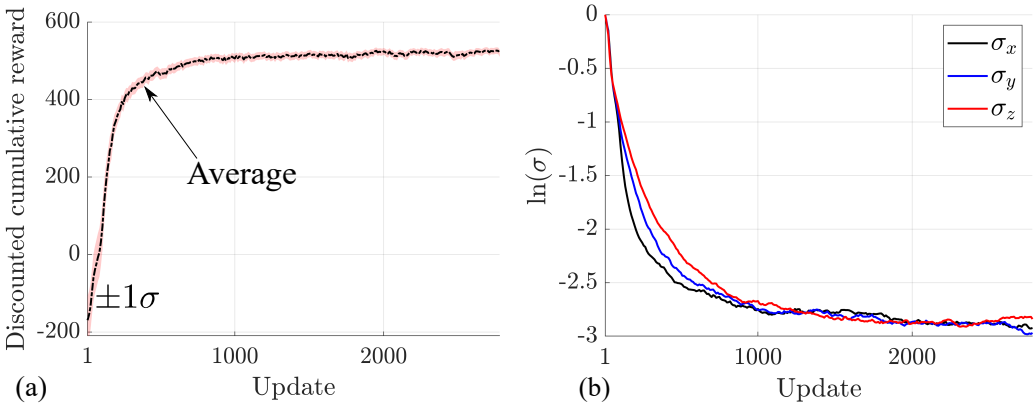

Figure 3. Summary of the training process for long-term station-keeping in the CR3BP via (a) the average discounted cumulative reward in black, with $\pm 1\sigma$ in red and (b) standard deviations of the Gaussian distribution for sampling actions.

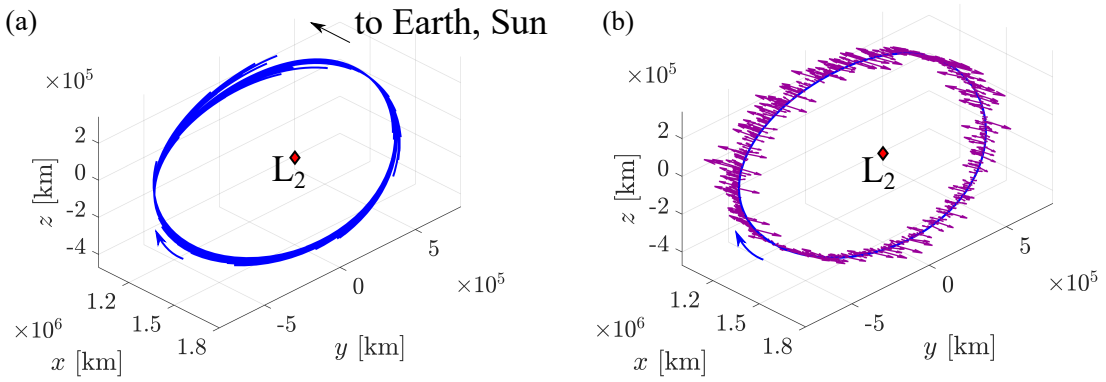

Figure 4. Evaluating the trained policy with 100 initial conditions near the reference halo orbit in the CR3BP: (a) associated trajectories propagated naturally and (b) controlled trajectories with station-keeping maneuvers.

This process is repeated for one revolution around the orbit. The resulting controlled trajectories are displayed in blue in Figure 4(b) and are visually indistinguishable from the reference at this scale. In fact, the maximum deviation from the reference across these 100 trajectories possesses a magnitude of 285 km and 0.13 m/s in position and velocity, respectively. Overlaid on this figure are purple arrows that supply a scaled representation of the station-keeping maneuvers generated by the trained policy. This figure demonstrates that the trained policy successfully achieves station-keeping over one orbital period for initial conditions defined close to any state along the reference.

Following training, the policy associated with long-term station-keeping near an L_2 southern halo orbit is evaluated in the presence of regular momentum unloads. In this scenario, a single initial condition is defined by applying a perturbation to a randomly selected location along the reference orbit. The policy is evaluated with this initial condition to produce an impulsive maneuver. Then, a momentum unload cycle is implemented using coast arcs in the natural CR3BP with a duration of $t_{MU} = 110$ hrs separated by momentum unloads instantaneously applied in a random direction with a magnitude of $\Delta v_{MU} = 8.7$ mm/s. The perturbed state at the beginning of the next time step is used to select the next station-keeping maneuver. This procedure is repeated for approximately 20 years, i.e., 40 revolutions of the reference orbit. For a single initial condition, the resulting controlled trajectory is displayed in blue in Figure 5(a), with purple arrows that depict the direction and relative magnitude of the station-keeping maneuvers. The magnitude of each of these station-keeping maneuvers is displayed in Figure 3(b). Over 20 years, 393 station-keeping maneuvers are performed with a cumulative maneuver magnitude of $\Delta v_{total} = 15.87$ m/s, well below the desired threshold of 5 m/s per year. Additionally, Figure 6 displays the time history of the magnitudes of the displacements in position and velocity, between the reference and controlled trajectory. The station-keeping maneuvers produce bounded motion near the reference halo orbit: over 20 years, the maximum distance from the reference is 390 km and the maximum difference in velocity possesses a magnitude of 0.24 m/s. Of course, these quantities may be reduced further with alternative hyperparameter values and neural network structures, additional training, or a nonperiodic reference. Nevertheless, the results presented in this example demonstrate the trained policy successfully designs maneuvers for long-term stationkeeping near an L_2 southern halo orbit in the Sun-Earth CR3BP with regular perturbations from momentum unloads.

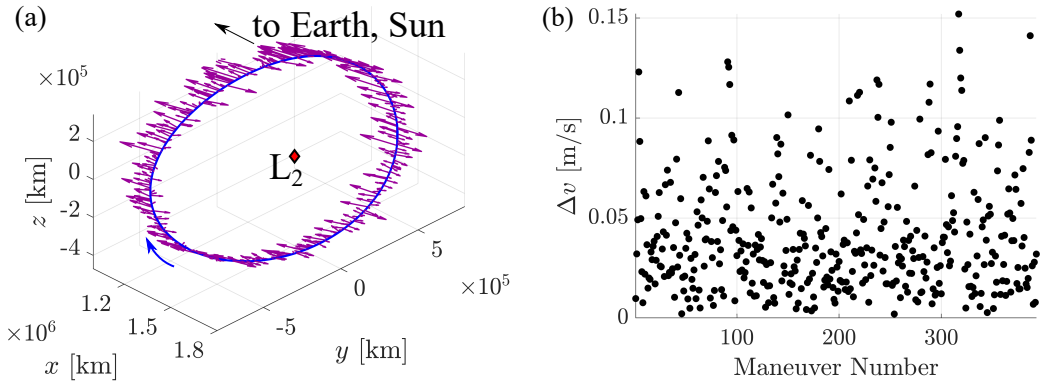

Figure 5. Station-keeping maneuvers in the CR3BP with regular momentum unloads: (a) single controlled trajectory propagated for 40 revolutions with maneuvers and (b) associated maneuver magnitudes.

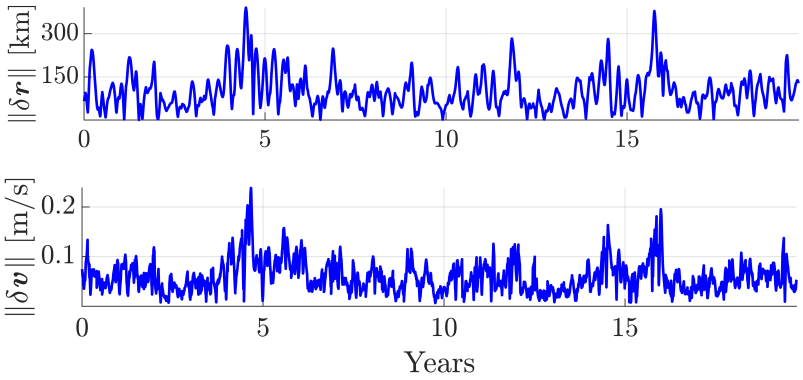

Figure 6. Time history of the magnitude of the position (top) and velocity (bottom) of the controlled trajectory in Figure 5 relative to the reference halo orbit.

6 RESULTS: STATION-KEEPING MANEUVER DESIGN IN AN EPHEMERIS MODEL

In this example, the RL maneuver planner is applied to a spacecraft operating near an L_2 quasi-halo orbit in the higher-fidelity Sun-Earth point mass ephemeris model. Recall from Section 4.3.3 that the RL problem formulation is similar to the previous example. The primary difference, however, is in the use of a nonperiodic, epoch-dependent reference trajectory and the incorporation of a time-like quantity in the definition of the observation vector. Using this problem formulation, the policy used to design station-keeping maneuvers is trained to follow the quasi-halo trajectory displayed in Figure 1 over the timespan $[29389.905, 32628.759]$ MJD, i.e., from 24 Jun 2021, 09:42:02 UTC to 7 May 2030, 06:11:48 UTC. Training continues for 3×10^6 time steps within the environment. This termination condition is defined using fewer time steps than the example presented in Section 5 due to the higher computational complexity associated with numerical integration in a point mass ephemeris model. However, the policy is effectively trained within this maximum bound on the number of time steps to converge on a solution that possesses a high discounted cumulative reward and with a small standard deviation in the Gaussian distribution used to sample the actions, as displayed in Figure 7 using a configuration that is consistent with Figure 3. Analysis of Figure 7(a) reveals that the discounted cumulative reward increases throughout the learning process and approaches a plateau, while the associated standard deviation decreases across updates. Fur-

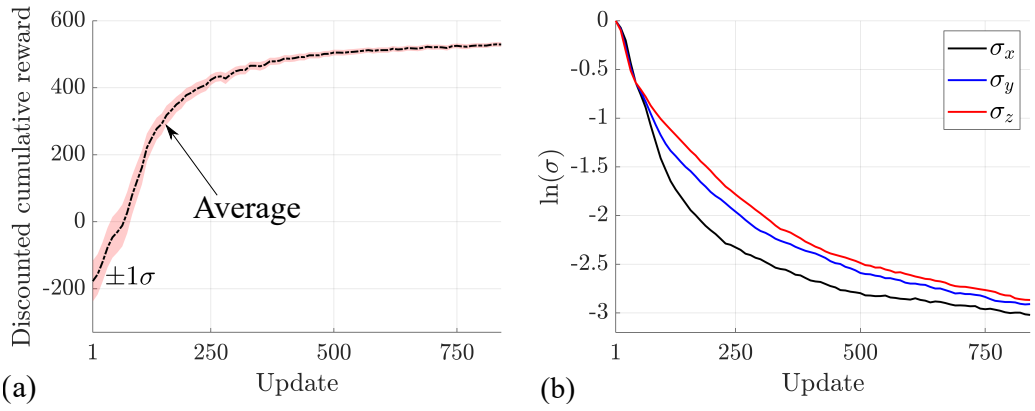

Figure 7. Summary of the training process for long-term station-keeping in the ephemeris model via (a) the average discounted cumulative reward in black, with $\pm 1\sigma$ in red and (b) standard deviations of the Gaussian distribution for sampling actions.

thermore, Figure 7(b) displays the natural logarithm of the standard deviations associated with the Gaussian distributions for sampling the actions across each update. Together, these two figures indicate a successful learning process that produces a policy that possesses both a high discounted cumulative reward and confidence in the associated actions.

The trained policy is evaluated to design station-keeping maneuvers for a spacecraft operating near an L_2 quasi-halo in a Sun-Earth point mass ephemeris model with regular momentum unloads perturbing the trajectory. In this example, one initial state for the spacecraft is defined by applying a perturbation from the reference path at an epoch of $t_0 = 29400$ MJD. The first station-keeping maneuver is selected by evaluating the policy. A momentum unload cycle is then implemented. The observation vector at the end of this momentum unload cycle supplies the observation vector at the beginning of the next time step that is used to evaluate the policy again. This process repeats for approximately 8.84 years. The resulting controlled trajectory is displayed in blue in Figure 8(a) in the Sun-Earth rotating frame with dimensional coordinate relative to the Earth with scaled maneuvers represented by purple arrows. To supplement this information, Figure 9 displays the time history of the magnitudes of the displacements in position and velocity between the reference and perturbed

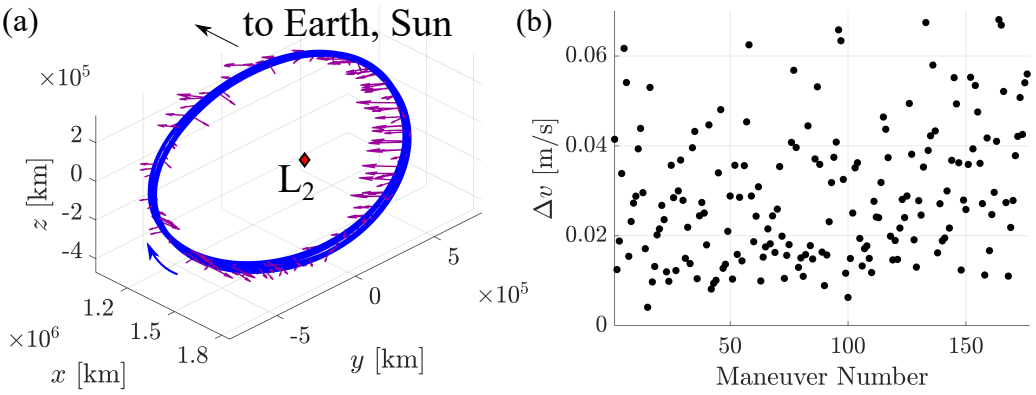

Figure 8. Station-keeping maneuvers in the ephemeris model with regular momentum unloads: (a) single controlled trajectory propagated for approximately 17.93 revolutions with maneuvers and (b) associated maneuver magnitudes.

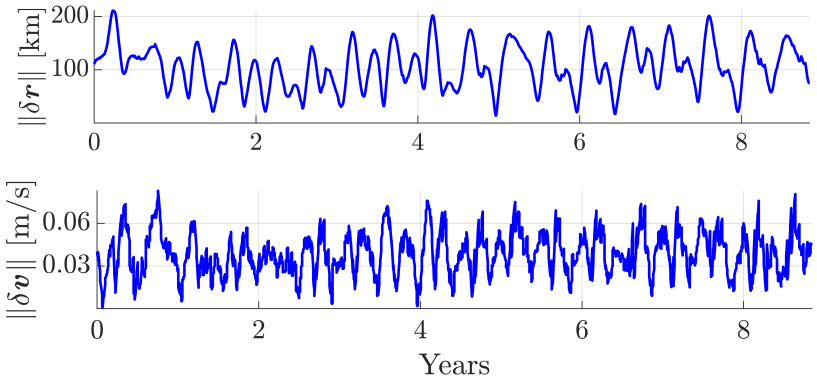

Figure 9. Time history of the magnitude of the position (top) and velocity (bottom) of the controlled trajectory in Figure 8 relative to the reference quasi-halo trajectory.

trajectories. Over 8.84 years, the controlled trajectory remains within 211 km and 0.08 m/s of the reference, even in the presence of perturbations via momentum unloads. Figure 8(b) then displays the associated maneuver magnitudes: the 176 maneuvers that occur over these 8.84 years require a total budget of $\Delta v_{total} = 5.13$ m/s, well below the desired threshold of 5 m/s per year. This value of Δv_{total} is similar to that required over 8.84 years for the previous example, formulated in the CR3BP; small differences are likely due to the use of a nonperiodic reference path in this example as well as the distinct evaluation trajectories and recovered policies. Together, these results indicate that the RL implementation successfully produces a policy that generates station-keeping maneuvers for a spacecraft to remain near the L_2 quasi-halo trajectory in the point mass ephemeris model with a low required control effort, even as momentum unloads perturb the path. This proof of concept will be extended in future work to: 1) implement more complex maneuvering goals, 2) incorporate constraints, 3) incorporate additional perturbations, and 4) train policies for a wider array of reference paths and higher-fidelity environments.

7 CONCLUSIONS

Reinforcement learning (RL) is used to design station-keeping maneuvers for a spacecraft operating near a Sun-Earth L_2 southern halo orbit. First, the goal of long-term station-keeping maneuver design to cumulatively minimize both the deviation from a reference path and the required control effort is translated into an RL problem. Then, Proximal Policy Optimization is used to train policies that design these maneuvers in two environments of increasing complexity: the Sun-Earth circular restricted three-body problem (CR3BP) and a Sun-Earth point mass ephemeris model. Hyperparameters governing PPO and the neural network structure are selected using Bayesian optimization. Then, the maneuvers recovered through the RL implementation are verified in a simplified greedy maneuver design scenario via comparison to expected results from dynamical systems theory.

The constructed RL-based maneuver is applied to long-term station-keeping maneuver design in each of the two environments. In these examples, the RL-based maneuver planner is successfully trained in each dynamical environment to converge on a policy with a high discounted cumulative reward and high confidence in the corresponding actions. The two trained policies are then evaluated with regular perturbations due to momentum unloads. Evaluating the scenario formulated in the CR3BP with a single perturbed initial condition, a total maneuver magnitude of 15.87 m/s is required over 20 years for the spacecraft to possess a position and velocity that remains within 390 km

and 0.24 m/s, respectively, of the reference southern halo orbit. In the point mass ephemeris model, the second trained policy is evaluated with a single perturbed initial condition. Over the span of 8.84 years, a total maneuver magnitude of 5.13 m/s is required for the spacecraft to remain within 211 km and 0.08 m/s of the reference quasi-halo trajectory. In both the low- and high-fidelity dynamical environments, the RL implementation successfully designs maneuvers that balance minimizing long-term boundedness to the vicinity of a reference with minimizing control effort. Motivated by these results, an alternative paradigm to station-keeping maneuver design that leverages RL offers a foundation for continued development of autonomous maneuver planning capabilities that will support future missions operating within multi-body regimes throughout our solar system.

8 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

[1] N. Bosanac, C.M. Webster, K.C. Howell, D.C. Folta, "Trajectory Design for the Wide Field Infrared Survey Telescope Mission," *Journal of Guidance, Control and Dynamics*, Vol. 42, No. 9, September 2019, pp. 1989 – 1911, doi:10.2514/1.G004179.

[2] D. Folta and M. Beckman, "Libration Orbit Mission Design: Applications of Numerical and Dynamical Analysis," *Libration Orbit Mission Design: Applications of Numerical and Dynamical Analysis, Aiguablava, Spain*, June 2002.

[3] C. Simó, G. Gómez, J. Llibre, R. Martinez, and L. Rodriguez, "On the Optimal Station Keeping Control of Halo Orbits," *Acta Astronautica*, Vol. 15, No. 6, 1987, pp. 391–397, doi:10.1016/0094-5765(87)90175-5.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, Vol. 518, 02 2015, pp. 529–33, doi:10.1038/nature14236.

[5] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," *arXiv preprint arXiv:1602.01783v2*, 2016.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, Vol. 529, 2016, pp. 484–489.

[7] A. Das-Stuart, K.C. Howell, D.C. Folta, "Rapid Trajectory Design in Complex Environments Enabled by Reinforcement Learning and Graph Search Strategies," *Acta Astronautica*, Vol. 171, 2020, pp. 172–195.

[8] D. Miller, R. Linares, "Low-Thrust Optimal Control via Reinforcement Learning," *29th AAS/AIAA Space Flight Mechanics Meeting, Ka'anapali, HI*, 2019.

[9] C.J. Sullivan, N. Bosanac, "Using Multi-Objective Deep Reinforcement Learning to Uncover a Pareto Front in Multi-Body Trajectory Design," *AAS/AIAA Astrodynamics Specialist Conference*, (Virtual) South Lake Tahoe, CA, August 9-13, 2020.

[10] C.J. Sullivan, N. Bosanac, "Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System," *30th AIAA/AAS Space Flight Mechanics Meeting, Orlando, FL*, 2020.

[11] N. LaFarge, D. Miller, K.C. Howell, R., Linares, "Guidance for Closed-Loop Transfers using Reinforcement Learning with Application to Libration Point Orbits," *30th AIAA/AAS Space Flight Mechanics Meeting, Orlando, FL*, 2020, doi:10.2514/6.2020-0458.

[12] A. Scorsoglio, R. Furfaro, R. Linares, M. Massari, "Actor-Critic Reinforcement Learning Approach to Relative Motion Guidance in Near-Rectilinear Orbit," *29th AAS/AIAA Space Flight Mechanics Meeting, Ka'anapali, HI*, 2019.

[13] D. Guzzetti, "Reinforcement Learning and Topology of Orbit Manifolds for Stationkeeping of Unstable Symmetric Periodic Orbits," *AAS/AIAA Astrodynamics Specialist Conference, Portland, ME*, July 2019.

[14] A. Molnar, "Hybrid Station-Keeping Controller Design Leveraging Floquet Mode and Reinforcement Learning Approaches," Master's thesis, School of Aeronautics Astronautics, Purdue University, West Lafayette, IN, December 2020.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv prePrint arXiv:1912.01703*, 2019.

[16] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. London, UK: Academic Press, 1967.

[17] "SPICE Toolkit," `https://naif.jpl.nasa.gov/naif/aboutspice.html`, 2016. Accessed December 2020.

[18] T. Pavlak, *Trajectory Design and Orbit Maintenance Strategies in Multi-body Dynamical Regimes*. PhD thesis, School of Aeronautics Astronautics, Purdue University, West Lafayette, IN, May 2013.

[19] V.R. Konda, J.N. Tsitsiklis, "Actor-Critic Algorithms," *Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Denver, CO*, 2000, pp. 1008–1014.

[20] S.P. Singh, T.S. Jaakkola, M.I. Jordan, "Learning Without State-Estimation in Partially Observable Markovian Decision Processes," *ICML*, 1994.

[21] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, O. Bachem, "What Matters in On-Policy Reinforcement Learning? A Large-Scale Empirical Study," *Google Research, Brain Team*, 2020, arXiv:2006.05990.

[22] J. Schulman, Wolski, P. Dhariwal, A. Radford, O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.

[23] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger, "Deep Reinforcement Learning that Matters," *Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA*, 2018.

[24] K.Hornik, M. Stinchcombe, H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.

[25] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," *arXiv:1811.03378v1*, 2018.

[26] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," *Neural Networks for Perception*, pp. 65–93, Elsevier, 1992.

[27] D. P. Bertsekas, *Reinforcement learning and optimal control*. Belmont, MA: Athena Scientific, 2019.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, San Diego, CA*, 2015.

[29] M. Sewak, *Deep Reinforcement Learning - Frontiers of Artificial Intelligence*. Springer, 2019.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[31] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[32] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[33] Z. Zhang, X. Luo, T. Liu, S. Xie, J. Wang, W. Wang, Y. Li, and Y. Peng, "Proximal Policy Optimization with Mixed Distributed Training," *2019 IEEE 31st International Conference on Tools with Artificial Intelligence*, 2019, pp. 1452–1456.

[34] M.S. Holubar, M.A. Wiering, "Continuous-Action Reinforcement Learning for Playing Racing Games: Comparing SPG to PPO," *arXiv:2001.05270*, 2020.

[35] T.A. Pavlak, K.C. Howell, "Strategy for Optimal, Long-Term Stationkeeping of Libration Point Orbits in the Earth-Moon System," *AIAA/AAS Astrodynamics Specialist Conference*, Minneapolis, MN, August 13-16, 2012, doi:10.2514/6.2012-4665.

[36] A. Farrés, G. Gómez, J.J. Masdemont, C. Webster, D.C. Folta, "The Geometry of Stationkeeping Strategies around Libration Point Orbits," *70th International Astronautical Congress*, Washington DC, 2019.

[37] M.T. Young, J.D. Hinkle, R. Kannan, A. Ramanathan, "Distributed Bayesian optimization of deep reinforcement learning algorithms," *Journal of Parallel and Distributed Computing*, Vol. 139, May 2020, pp. 43–52, doi:10.1016/j.jpdc.2019.07.008.

[38] H.S. Jomaa, J. Grabocka, L. Schmidt-Thieme, "Hyp-RL : Hyperparameter Optimization by Reinforcement Learning," *arXiv:1906.11527v1*, 2019.

[39] E. Brochu, V.M. Cora, N. de Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning," *University of British Columbia Department of Computer Science*, December 2010, http://arXiv.org/abs/1012.2599.

[40] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014. Accessed November 2020.