DEEP REINFORCEMENT LEARNING FOR STATION KEEPING ON NEAR RECTILINEAR HALO ORBITS

Suda, Takumi; Shimane, Yuri; Elango, Purnanand; Weiss, Avishai

TR2023-098 August 18, 2023

Abstract

In this work, we develop and evaluate a soft actor-critic (SAC) deep reinforcement learning (DRL) policy for station keeping of a spacecraft on a near-rectilinear halo orbit (NRHO) in the full-ephemeris dynamics. Monte Carlo simulations show that the DRL-based NRHO station-keeping policy maintains an approximately linear increase in delta-v at the apolune of each revolution, with a low spread in the delta-v gradient across the samples

AIAA/AAS Astrodynamics Specialist Conference 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

(Preprint) AAS 23-401

DEEP REINFORCEMENT LEARNING FOR STATION KEEPING ON NEAR RECTILINEAR HALO ORBITS

Takumi Suda[*], Yuri Shimane[†], Purnanand Elango[‡], Avishai Weiss[§]

In this work, we develop and evaluate a soft actor-critic (SAC) deep reinforcement learning (DRL) policy for station keeping of a spacecraft on a near-rectilinear halo orbit (NRHO) in the full-ephemeris dynamics. Monte Carlo simulations show that the DRL-based NRHO station-keeping policy maintains an approximately linear increase in delta-v at the apolune of each revolution, with a low spread in the delta-v gradient across the samples.

INTRODUCTION

The Lunar Orbital Platform-Gateway (LOP-G) will serve as an outpost orbiting the Moon, allowing for communication with both the lunar surface and Earth. The Gateway will fly on a 9:2 resonant near-rectilinear halo orbit (NRHO) around the Earth-Moon L_2 [1, 2], which means that this orbit revolves around the Moon nine times for every two revolutions of the Moon around the Earth. This orbit was selected for the Gateway since it can avoid solar eclipses by the Earth and the Moon, and spends much of its time over the southern hemisphere of the Moon, where a future lunar base is to be established. When considering perturbations such as the Sun's gravitational attraction, ephemeris-based planetary positions, solar radiation pressure, and lunar J2 gravitational harmonics, the NRHO is an unstable and aperiodic trajectory. This trajectory is known as a long-horizon (LH) reference orbit, and can be computed, e.g., via multiple shooting [3] and forward-backward shooting using the sparse solver SNOPT [4]. Due to navigational uncertainty, the Gateway will require station-keeping maneuvers to maintain the LH reference orbit and prevent rapid divergence.

In recent years, several methodologies have been proposed for maintaining a spacecraft on an NRHO. Spectrum-based strategies align a spacecraft with the stable subspace near the reference trajectory [5–7]; for example, Cauchy-Green Tensor (CGT) targeting [8–11] utilizes the eigendecomposition of the CGT to bring the spacecraft towards a path contracting to the reference trajectory. In contrast, target point approaches control the spacecraft state or a portion of the state at some future time to be some desired value, see e.g., xz-plane crossing control [8, 10–15] in which the spacecraft can be kept on the orbit by maintaining the symmetry of the orbit along the line of the two primary attractors.

Meanwhile, machine learning technology has made significant strides in recent decades, including in the area of optimal control via deep reinforcement learning (DRL). One key advantage of DRL is its efficient use of computational resources. Once the neural-network-based policy has been

[*] Software Engineer, Communication Systems Center, Mitsubishi Electric Corporation, Hyogo, Japan.

[†] Ph.D. Candidate, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

[‡] Ph.D. Candidate, Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195.

[§] Senior Principal Research Scientist, Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139.

trained, it can be evaluated to produce an optimal control without requiring additional computations, which may be attractive for on-board implementation. Additionally, DRL provides the flexibility to customize the reward function to maximize or minimize specific objectives. Finally, unlike xz-plane crossing control or CGT targeting, which are heuristic and rely on exploiting certain features of halo orbits, DRL can search for the optimal solution without depending on any specific characteristics of the dynamics. As such, DRL has been applied to station keeping and relative motion on ephemeris-based multi-body dynamics [16–22]. The works in [16, 18] seek to calculate the optimal maneuver for on-board implementation. DRL has been applied for multiple spacecraft rendezvous on halo orbits in [20, 21]. In [19], the focus was on speeding up the training process. DRL has also been used for low-thrust station-keeping control in [17, 22]. LaFarge et al. [22] consider the use of a DRL policy as a seeding scheme to classical differential correction in order to obtain an efficient and robust low-thrust control scheme; this is applied to an experiment aimed at bringing the spacecraft from a diverging path back to the nominal NRHO, where a traditional station-keeping approach is insufficient.

In this work, we consider the scenario where a mission is to be flown on an NRHO, and an efficient station-keeping scheme specifically tailored for this mission is sought. As such, instead of relying on heuristic control schemes that are "myopic" in the sense that they only use information of the instantaneous state on the NRHO as well as some future perlinues, we aim to obtain a controller that can leverage the time-span of the non-autonomous, full-ephemeris dynamics specific to this hypothesized mission, in order to extract the maximum achievable performance in terms of station-keeping cost. While under perfectly known deterministic dynamics and perfect state knowledge, this could be posed as an open-loop trajectory optimization problem, the more realistic case involving imperfect state knowledge and/or an imperfect model of the dynamics poses a challenge to traditional optimization or optimal control approaches. To this end, we study the use of a DRL-based station-keeping policy for a spacecraft on an NRHO using the Soft Actor-Critic (SAC) [23] scheme. The DRL neural network is designed in a way that is tailored to a given mission, incorporating inputs and a value function that draw from the xz-plane crossing control scheme.

The remainder of this paper is organized as follows. First, we introduce the dynamical system model, the construction of a baseline NRHO, along with a review of the heuristic xz-plane crossing control. We then provide details on the proposed use of a DRL scheme for NRHO station keeping. This is followed by results on numerical experiments involving the DRL agent as well as the xz-plane crossing control. Finally, future work is discussed and concluding remarks are provided.

DYNAMICAL SYSTEM MODEL

This section provides a review on the dynamical system model employed in this work. A discussion on the xz-plane crossing control, which is a well-known station-keeping scheme for NRHOs, is also introduced, in order to provide a comparison to the DRL approach proposed in the subsequent section.

N-body Equations of Motion

The spacecraft dynamics are modeled with N-body equations of motion that include the Moon, the Earth, and the Sun, and are perturbed by the J2 spherical harmonics term of the Moon and solar radiation pressure. This model is chosen based on a prior study on the important perturbing accelerations for a spacecraft on a NRHO [12]. Let $\theta = [\boldsymbol{r}^T, \boldsymbol{v}^T]^T \in \mathbb{R}^6$ be the position and velocity state of the spacecraft; its natural dynamics is given by

$$\begin{aligned}\dot{\boldsymbol{r}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{v}} &= -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_{\mathrm{J2}} + \sum_i \boldsymbol{a}_{N_i} + \boldsymbol{a}_{\mathrm{SRP}},\end{aligned} \tag{1}$$

where $\boldsymbol{r}$ is the position vector of the spacecraft with respect to the Moon in an inertial frame, and the last three terms are, in order, the J2 perturbation of the Moon, the third-body perturbations of the Earth and Sun, and the solar radiation pressure. The ephemerides of the planets are taken from NAIF's SPICE toolkit [24]. The expression for the perturbing accelerations can be found in [25].

Baseline NRHO in the Full-Ephemeris Model

A LH reference orbit, or *baseline NRHO*, is computed as a solution to (1), and is denoted as $\hat{\theta}(t)$. The baseline represents the nominal zero-effort path that the spacecraft is to follow. To generate the baseline, we start with a desired periodic orbit in the circular restricted three-body problem, and "stack" it on itself to form a multi-period initial guess to transition into the full-ephemeris model. The transition can be done effectively with multiple shooting and is well-documented in the literature [26]. Since the dynamics about the baseline trajectory are unstable, station-keeping maneuvers are required.

State Uncertainty and Control Action

The control action is assumed to be applied as an impulsive change in the velocity vector $\Delta\boldsymbol{v}$, such that at the maneuver epoch t,

$$\boldsymbol{v}(t^+) = \boldsymbol{v}(t^-) + \Delta\boldsymbol{v}. \tag{2}$$

For realistic station keeping, the spacecraft only has access to a noisy state $\tilde{\theta}$ that mimics the output of an on-board navigation filter, and is given by

$$\tilde{\theta} = \theta + \begin{bmatrix}\boldsymbol{e}_r \\ \boldsymbol{e}_v\end{bmatrix}, \tag{3}$$

where $\boldsymbol{e}_r$ and $\boldsymbol{e}_v$ are position and velocity errors, modeled as zero-mean Gaussian random variables with standard deviations σ_r and σ_v.

Review of xz-plane Crossing Control Strategy

The DRL-based station keeping adopted in this work is akin to the xz-plane crossing control, which is a well-known strategy for stabilizing a spacecraft on a NRHO [8]. We provide a brief review of xz-plane crossing control, as well as an overview of the controller implementation. The xz-plane crossing controller will be compared against the DRL-based station-keeping scheme that we propose in the subsequent section.

The key idea behind the xz-plane controller is based on the observation that halo orbits in restricted three-body models exhibit a symmetry about the xz-plane in the Earth-Moon rotating frame. This necessitates that the spacecraft state at perilune and apolune of an NRHO lies on this plane, and has a velocity vector that is perpendicular to this plane. While this condition does not exactly hold in the full-ephemeris model, previous investigations have shown that the spacecraft may be kept on its quasi-periodic path if the spacecraft's velocity vector is kept close to perpendicular at subsequent xz-plane crossings [8, 11].

Algorithm 1, adopted from [7], shows the implementation of the baseline-relative xz-plane controller implemented in this work. At a given control maneuver instance, the algorithm requires the current epoch t_0, current state estimate $\tilde{\theta}(t_0)$, the ephemeris information of the baseline $\hat{\theta}(t)$, a scalar N denoting the future perilune to be targeted, a tolerance on the targeting violation ϵ, and a maximum number of iterations for the algorithm. The algorithm then starts by computing the time it takes for the current estimated state to arrive at its N^{th} perilune into the future, t_p, and querying the baseline state at the N^{th} perilune into the future, $\hat{\theta}_p$. While in this work perilune is used as the targeting event, it may be replaced by the instance of xz-plane crossing in the Earth-Moon rotating frame; whereas these two conditions are identical in the CR3BP, they do not correspond to the exact same epoch in a full-ephemeris model. Nevertheless, no significant difference in performance was observed from the choice of the targeting event.

Note also that the baseline state at the N^{th} future perilune is *not* the baseline state at epoch $t = t_0 + t_p$; denote the time taken by the baseline state at epoch t_0 to reach its N^{th} perilune as $\hat{t}_p$,

$$t_p \neq \hat{t}_p \to \hat{\theta}_p = \hat{\theta}(t_0 + \hat{t}_p) \neq \hat{\theta}(t_0 + t_p). \tag{4}$$

Thus, the targeting scheme works by comparing the future projection of the current estimated state against the baseline state at some future targeting event (e.g. N^{th} perilune), which will involve a temporal discrepancy.

The three custom functions in lines 1, 2, and 6/7 of Algorithm 1 are as follows:

- The function $\textsc{SearchPerilune}(\tilde{\theta}, N)$ solves an initial value problem (IVP) over $t \in [0, (N+1) \times P]$, where P is (approximate) period of the NRHO, and returns the time t_p it takes for the state $\tilde{\theta}$ to reach the N^{th} future perilune. Note that by propagating over $N+1$ periods into the future, the N^{th} perilune will occur before the final time of the IVP.
- The function $\textsc{QueryPerilune}(\hat{\theta}, t_0, N)$ queries the N^{th} future perilune from the epoch t_0 of the baseline $\hat{\theta}$. This can for example be a simple query of a look-up table of the epoch and perilune states of the baseline that can be prepeared ahead of time.
- The function $\textsc{InertialToEarthMoonRotating}(\cdot, t)$ applies the transformation from the inertial to the Earth-Moon rotating frame on the state or the state transition matrix. Note that the function also necessitates an epoch t, since the Earth-Moon rotating frame in the full-ephemeris model is epoch-dependent.

The for-loop starting on line 4 computes the required ΔV vector via linearization, and a break clause is added to check if the projected future perilune v_x has a difference in magnitude compared to the baseline's corresponding future perilune v_x that is below a threshold ϵ. With a threshold $\epsilon = 20$ m/s, the algorithm is found to converge consistently within a few iterations, and thus a maximum number of iterations of 10 for the for-loop is found to be sufficient. The computation of the incremental $\Delta \boldsymbol{v}$ vector in line 11 has a closed form solution based the the projection of the origin in $\mathbb{R}^3$ onto the intersection of two half-spaces, see [7] for detail. The additional multiplier $\eta > 0, \eta \approx 0$ in line 11 is to ensure the numerical solution of the projection problem satisfies the break condition in line 8.

Algorithm 1 Baseline-relative xz-plane control

```
Input: $t_0, \tilde{\theta}(t_0), \hat{\theta}(t), N, \epsilon$, maxiter
$t_{\mathrm{p}} \leftarrow$ SearchPerilune$(\tilde{\theta}, N)$    ▷ Compute time to $N^{\mathrm{th}}$ future perilune along state estimate
$\hat{\theta}_p \leftarrow$ QueryPerilune$(\hat{\theta}, t_0, N)$    ▷ Query baseline state on $N^{\mathrm{th}}$ future perilune in Earth-Moon rotating frame
$\Delta \boldsymbol{v} \leftarrow [0\ 0\ 0]^{\top}$    ▷ Initialize $\Delta V$ vector
for $i$ in maxiter do
    $\tilde{\theta}_p, \Phi_p \leftarrow$ SolveIVP$(\tilde{\theta}, t_{\mathrm{p}})$    ▷ Propagate current state estimate and state transition matrix until target perilune time
    $\dot{x}_{\mathrm{p}} \leftarrow$ InertialToEarthMoonRotating$(\tilde{\theta}_p, t_0 + t_p)$    ▷ Convert state from inertial frame to Earth-Moon rotating frame and keep only $v_x$
    $\Phi_P \leftarrow$ InertialToEarthMoonRotating$(\Phi_p, t_0 + t_p)$    ▷ Convert state transition matrix from inertial frame to Earth-Moon rotating frame
    if $|\dot{x}_{\mathrm{p}} - \hat{\theta}_p^{[4]}| \leq \epsilon$ then
        break    ▷ Converged
    end if
    $\Delta \boldsymbol{v}_{\mathrm{increment}} \leftarrow \underset{y \in \mathbb{R}^3}{\mathrm{argmin}} \|y\|_2$ subject to $|\Phi_{\mathrm{p}}^{[4,4:6]} y + (\dot{x}_{\mathrm{p}} - \hat{\theta}_p^{[4]})| \leq (1-\eta)\epsilon$    ▷ Compute minimum-norm delta-v as projection on intersection of two half-spaces
    $\Delta \boldsymbol{v} \leftarrow \Delta \boldsymbol{v} + \Delta \boldsymbol{v}_{\mathrm{increment}}$    ▷ Update delta-v
end for
```

DEEP REINFORCEMENT LEARNING FOR NRHO STATION KEEPING

The decision-making process of the DRL is based on a Markov decision process (MDP), which relies solely on the current state to determine the future state. Figure 1 displays the neural network architecture of the DRL. We used Proximal Policy Optimization (PPO) [27] as the reinforcement learning algorithm to generate the control policy. PPO can effectively learn and adapt to new tasks in a stable manner, making it a suitable choice for our application [28].

This work considers the applicability of a DRL agent to perform station keeping assuming a pre-defined mission and a corresponding baseline NRHO. As such, the training environment as well as the network inputs are selected in such a way that the agent is able to leverage the variability of the dynamics specifically over this pre-defined mission. While this gives the DRL agent the potential to adapt to the specific time-window and out-perform heuristic-based control strategies such as the xz-plane controller described in the previous section, this also renders the DRL policy inadequate to be used for other epochs or other phasing locations along the NRHO. If, in contrast to our aim, a generalized DRL policy for performing station-keeping on any arbitrary baseline NRHO at any epoch is needed, the training data-set and the inputs must be modified from what will be described in this section.

Training and Testing Environment

The training and testing environment consists of a spacecraft that is inserted into a pre-computed baseline NRHO with an initial state uncertainty based on insertion standard deviations for position and velocity, σ_{r_i} and σ_{v_i}. The DRL agent is then tasked to compute a station-keeping maneuver at each revolution, given information that is computed based on a noisy state estimate $\tilde{\theta}$. The inputs to the network will be detailed further in the subsequent subsection. The ΔV maneuver hypothesized from the DRL is lower bounded by $\Delta V_{\min}$ component-wise. The reward function is based on the station-keeping cost at each action chosen by the actor; if the spacecraft diverges, a large penalty is applied.

Network Architecture

In this work, a relatively "shallow" deep neural network with two hidden layers is employed. The network uses the hyperbolic tangent $tanh$ as the activation function. Table 1 provides a summary of the network architecture, and a schematic is provided in Figure 1. We employ the Soft Actor-Critic (SAC) [23] paradigm, which is regarded as one of the most efficient RL algorithm to date, simultaneously tackling the two common issues in RL: sample inefficiency arising from on-policy learning, and brittleness coming from the sensitivity to the hyperparameters. Specifically, SAC is sample efficient as it is off-policy, while it has a reduced sensitivity to hyperparameters through an augmented value function that attempts to maximize the lifetime reward and the entropy of the policy.

Inputs Let the augmented state of the spacecraft $\theta_{\text{aug}} \in \mathbb{R}^7$ comprise of the position and velocity vectors and the elapsed mission time t. The inputs to both the actor and critic networks consists of the augmented state estimates at the next 7 xz-plane crossings, $\tilde{\theta}_{\text{aug}}(t_{p,i})$, where $i = 0, 1, \dots, 6$ denotes the i^{th} future perilune, as well as the difference in the estimate and baseline augmented states at the next 7 xz-plane crossings,

$$\tilde{\theta}_{\text{aug}}(t_{p,i}) - \hat{\theta}_{\text{aug}}(\hat{t}_{p,i}) = \begin{bmatrix} \tilde{\theta}(t_{p,i}) \\ t_{p,i} \end{bmatrix} - \begin{bmatrix} \hat{\theta}(\hat{t}_{p,i}) \\ \hat{t}_{p,i} \end{bmatrix}, \tag{5}$$

resulting in $7 \times 7 + 7 \times 7 = 98$ inputs to the network.

The inclusion of t as part of the augmented state makes the network tailored specifically to the pre-defined range of epoch along the baseline NRHO. As previously mentioned, this design choice makes the network unable to perform at epochs other than the trained range. Instead, this allows the network to learn the specific non-autonomous dynamics over the prescribed range, thus potentially reducing the station-keeping cost across this mission timeline. Furthermore, by using the augmented state difference from expression (5) as input, the network is made explicitly aware of the temporal discrepancy at the future perilunes.

Outputs The actor network output consists of the control action, given by the ΔV vector in $\mathbb{R}^3$, as well as the fire angle ϕ, which is defined based on the projection of the spacecraft's position vector to the yz-plane in in the Earth-Moon rotating frame,

$$\phi = -\operatorname{atan2}(z^{(EM)}, y^{(EM)}), \tag{6}$$

where $y^{(EM)}$ and $z^{(EM)}$ are resolved in the Earth-Moon rotating frame. The negative sign in front of ϕ is to define $\dot{\phi}$ to be positive clockwise, along the direction of motion of the NRHO. Each component of the ΔV vector is normalized such that the resulting action is bounded within -1 and

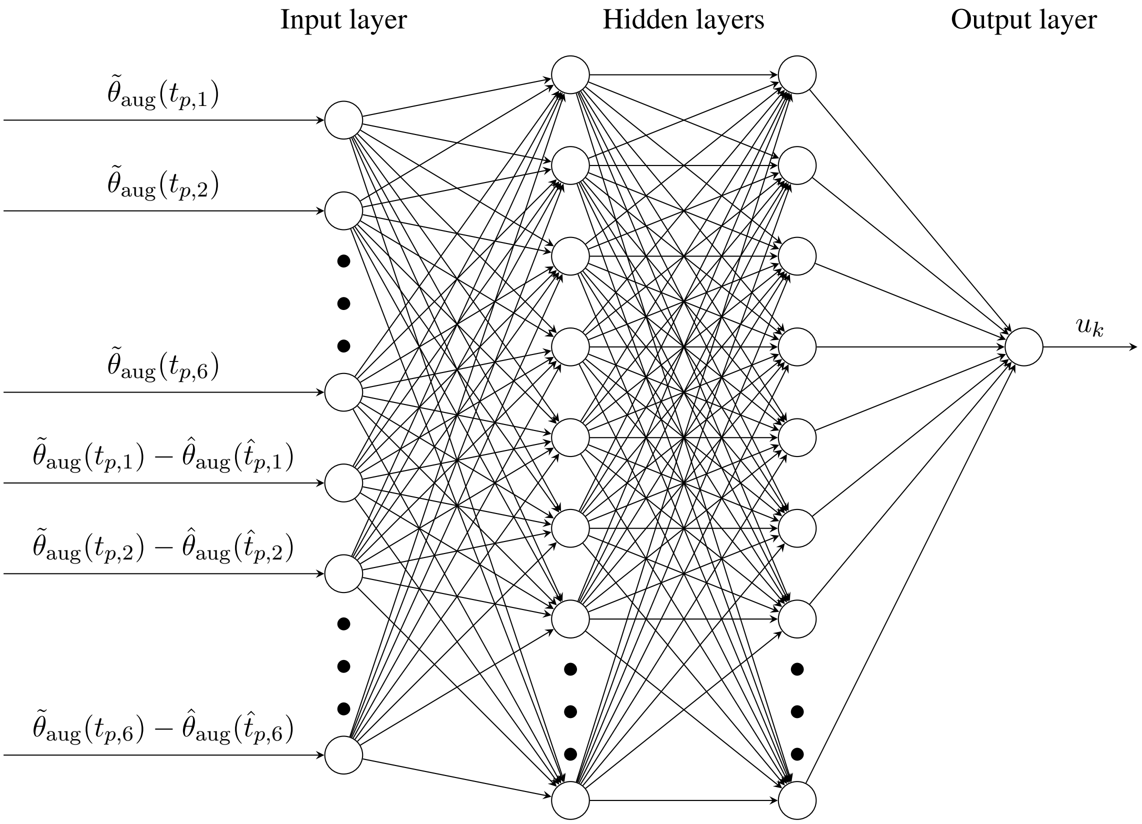

Input layer
Hidden layers
Output layer
$\tilde{\theta}_{\text{aug}}(t_{p,1})$
$\tilde{\theta}_{\text{aug}}(t_{p,2})$
$\tilde{\theta}_{\text{aug}}(t_{p,6})$
$\tilde{\theta}_{\text{aug}}(t_{p,1}) - \hat{\theta}_{\text{aug}}(\hat{t}_{p,1})$
$\tilde{\theta}_{\text{aug}}(t_{p,2}) - \hat{\theta}_{\text{aug}}(\hat{t}_{p,2})$
$\tilde{\theta}_{\text{aug}}(t_{p,6}) - \hat{\theta}_{\text{aug}}(\hat{t}_{p,6})$
u_k

Figure 1: Illustration of actor network architecture. Input consists of augmented state estimates at future perilunes, $\tilde{\theta}_{\text{aug}}(t_{p,k})$ as well as their difference from the baseline's corresponding perilune augmented states, $\tilde{\theta}_{\text{aug}}(t_{p,k}) - \hat{\theta}_{\text{aug}}(\hat{t}_{p,k})$.

$1\ \mathrm{m/s}$, and ϕ is bounded between 165° and 195°, where 180° corresponds to apolune. The value function of the critic network consists of the projected 28-revolution cumulative station-keeping cost.

NUMERICAL RESULTS

This section presents the training and testing results using the proposed DRL framework. The spacecraft is inserted into an NRHO at apolune at the initial state given in Table 2, with an orbit insertion error of 3-$\sigma_{r_i} = 30$ km and 3-$\sigma_{v_i} = 30$ cm/s. We assume navigation errors of 3-$\sigma_r = 1$ km and 3-$\sigma_r = 1$ cm/s, which is consistent with orbit determination that uses the Deep Space Network (DSN) [29]. The maneuver magnitude is upper-bounded to be 1 m/s per axis, resulting in an upper-bound ΔV magnitude of $\sqrt{3}$ m/s. The spacecraft equations of motion (1) are propagated using

Table 1: Number of neurons in each layer of actor and critic networks

Network	Input layer	1st hidden layer	2nd hidden layer	Output layer
Actor	98	488	244	4
Critic	98	488	244	1

the Runge-Kutta Prince-Dormand (8,9) integrator from the GNU GSL library [30]. The training is conducted using a batch size of 4000 samples, with up to 2.1×10^5 episodes. The trained network is then evaluated using 1000 test cases. The training and testing data both comprise of 28 NRHO revolutions, roughly corresponding to 6 months. This duration has been chosen as a compromise between computational cost and the number of navigation error realizations the agent sees within a single episode; with 28 revolutions, it is possible to estimate the annual cost consistently, while such an estimate may be inaccurate with a shorter simulation duration.

Table 2: Initial nominal state of the NRHO in the J2000 frame

Parameter	Initial value
Epoch, Julian date	2459957.50
x, km	-17762.62065600
y, km	33125.36549190
z, km	-60760.62704894
v_x, km/s	0.00086734
v_y, km/s	0.07118506
v_z, km/s	0.02308430

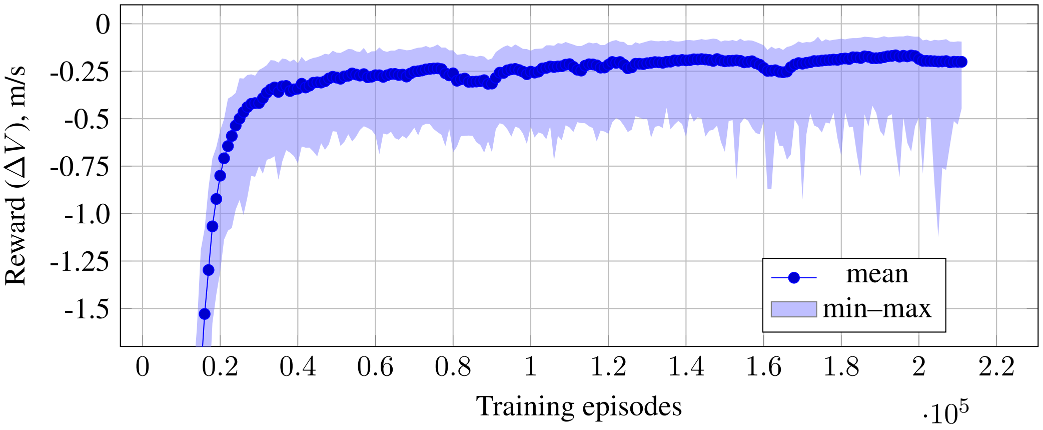


Figure 2: Reward against training episodes for the DRL training process

Performance over 28 NRHO Revolutions

First, the DRL agent's performance on the test data set is analyzed. Figures 3 and 4 show the performance of the DRL agent on the 1000 test cases. In Figure 3, the best, mean, and worst cumulative ΔV over 28 revolutions are shown. It is possible to see that in all three of these cases, the cost per revolution eventually settles to a relatively consistent maneuver cost per revolution. This consistency can be understood as the "steady-state" station-keeping performance, and is quantified by the gradient of the linear regression. Note that the linear regression excludes the costs arising from the first few revolutions; the cost over these first few revolutions may be understood as a "transient" cost, and is manifested on the larger variability of the y-intercept between the best/mean cases compared to the worst case. A large transient cost may be incurred when the orbital injection error is large. This variability of the transient cost is most apparent when comparing the histograms of the gradient and the y-intercept, as shown in Figure 4; the gradient over 56 revolutions has a

Table 3: Statistics of station-keeping ΔV for 1000 randomly chosen initial conditions

	Total over 28 revolutions with transient, cm/s	Transient, cm/s	Steady-state (gradient over 56 revolutions), cm/s
Min	8.8	1.8	13.1
Mean	20.4	12.2	22.8
Max	83.7	74.4	63.7

standard deviation of 4.37 cm/s, while the y-intercept has a standard deviation of 7.46 cm/s.

Table 3 gives the statistics of the transient and steady-state costs from the test cases. The worst performing case over 28 revolutions corresponds to a case with a very large transient cost of 74.4 cm/s, but with a gradient of 26.7 cm/s/(56 rev), which falls within a standard deviation from the mean gradient. We note that a worse gradient would have a larger impact as longer station keeping horizons are considered, thereby lessening the impact of the transient error.

We compare the results of the DRL agent's control with an implementation of xz-plane crossing control. We utilize the same baseline and navigation uncertainty as is used in the DRL environment, over the same 28 revolutions. In order to isolate the "steady-state" station keeping cost from the effect of insertion errors, for the xz-plane crossing control simulations we modify the orbit insertion error to be the same as the navigation uncertainty, i.e., 1 km and 1 cm/s. Figure 5 shows results for the cumulative ΔV over 56 revolutions, estimated by doubling the cumulative ΔV over 28 revolutions; the distribution has a mean of 165.89 cm/s and a standard deviation of 22.04 cm/s. This can be compared directly to Figure 4a. Note that the DRL agent performance bests the xz-plane crossing control in both consistency and magnitude, presumably due to the myopic (although generalizable) nature of xz-plane crossing control relative to the optimized DRL agent which extracts maximum performance on the particular baseline segment on which it is trained.

Sensitivity on Initial Insertion Error

The DRL policy is used to evaluate the sensitivity of the 28 revolution station-keeping cost on the insertion error in position and velocity. Figures 7 and 8 show the distribution of the initial insertion error in position and velocity spaces, respectively, for 500 randomly sampled initial insertion states, with the color-bar denoting the total station-keeping ΔV over 28 revolutions. As can be discerned from the figures (note the color-bar scale), the half-year station-keeping cost is more sensitive to 3-$\sigma_{v_i} = 30$ cm/s velocity errors than 3-$\sigma_{r_i} = 30$ km position errors. Additionally, a pattern in the station-keeping cost emerges under the random velocity errors shown in Figure 8, in that the cost to stabilize a spacecraft at apolune with off-nominal velocity to the NRHO seems to be more sensitive in x- ("out-of-plane") and y-directions ("velocity") than the z-direction ("radial") in the Earth-Moon rotating frame.

CONCLUSION

This work explored the use of a deep reinforcement learning (DRL) framework to plan the station-keeping maneuvers along a given baseline NRHO in order to achieve maximum performance in terms of station-keeping cost. The DRL network was specialized to a specific baseline NRHO through an active selection of input parameters, namely consisting of the future perilune state estimates and epochs, along with the deviation of these future perilune state estimates from the baseline

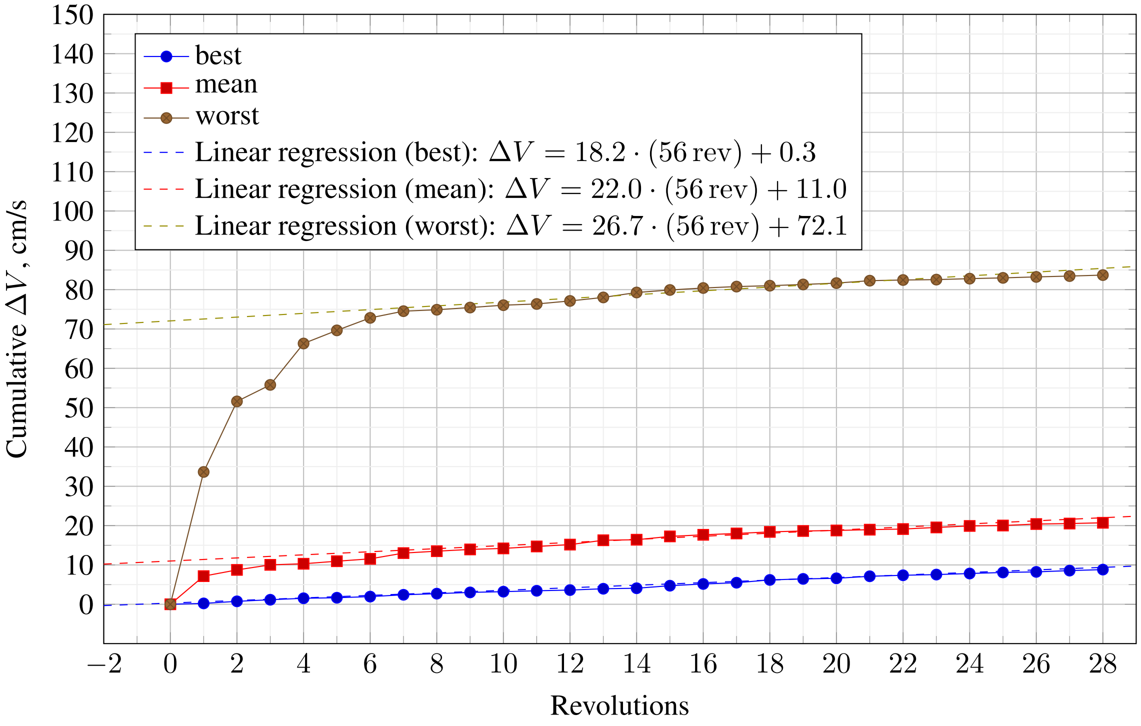
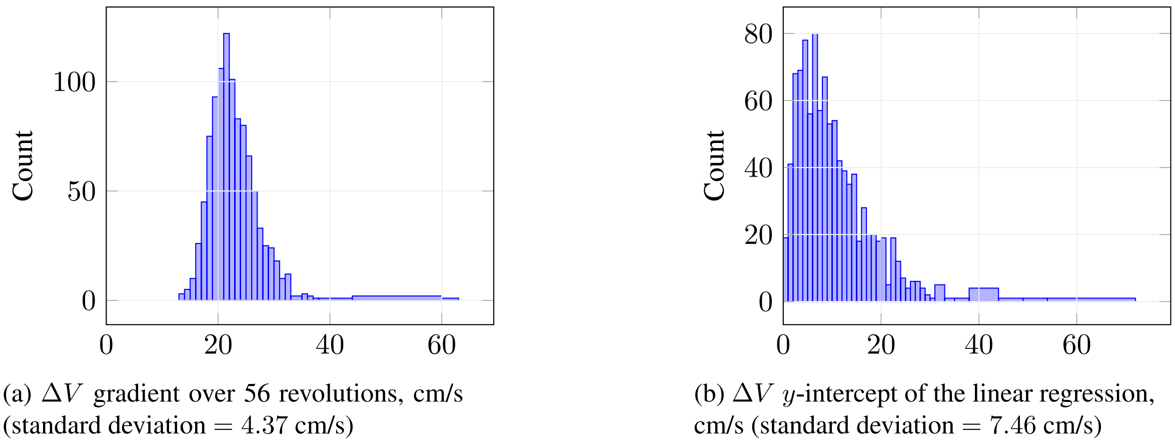


Figure 3: Cumulative ΔV over six months based on DRL agent, shown for the best, mean, and worst out of 1000 test samples. The best and worst regressions correspond to the test cases with least and most costly cumulative ΔV after 28 revolutions. The mean regression corresponds to the test case that results in the closest cumulative ΔV after 28 revolutions to the mean.



(a) ΔV gradient over 56 revolutions, cm/s (standard deviation = 4.37 cm/s)

(b) ΔV y-intercept of the linear regression, cm/s (standard deviation = 7.46 cm/s)

Figure 4: Histograms showing the spread of (a) the gradient and (b) y-intercept obtained for the linear regression based of the cumulative ΔV over 28 revolutions in 1000 samples. Note the gradient over 56 revolutions is simply 2 times the gradient over 28 revolutions.

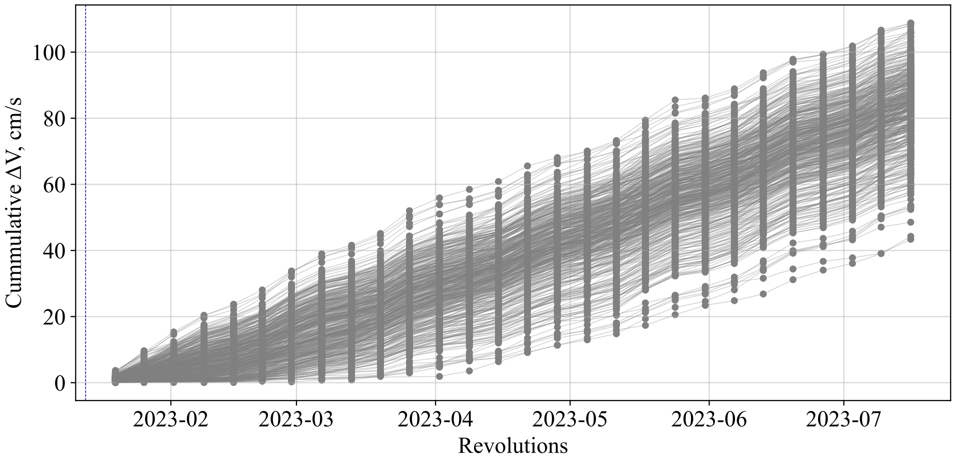


Figure 5: Cumulative ΔV over 28 revolutions from 500 Monte-Carlo samples with xz-plane crossing control

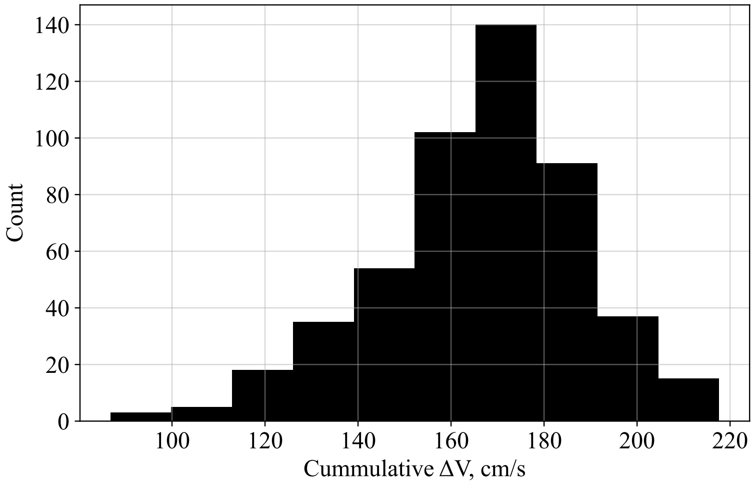
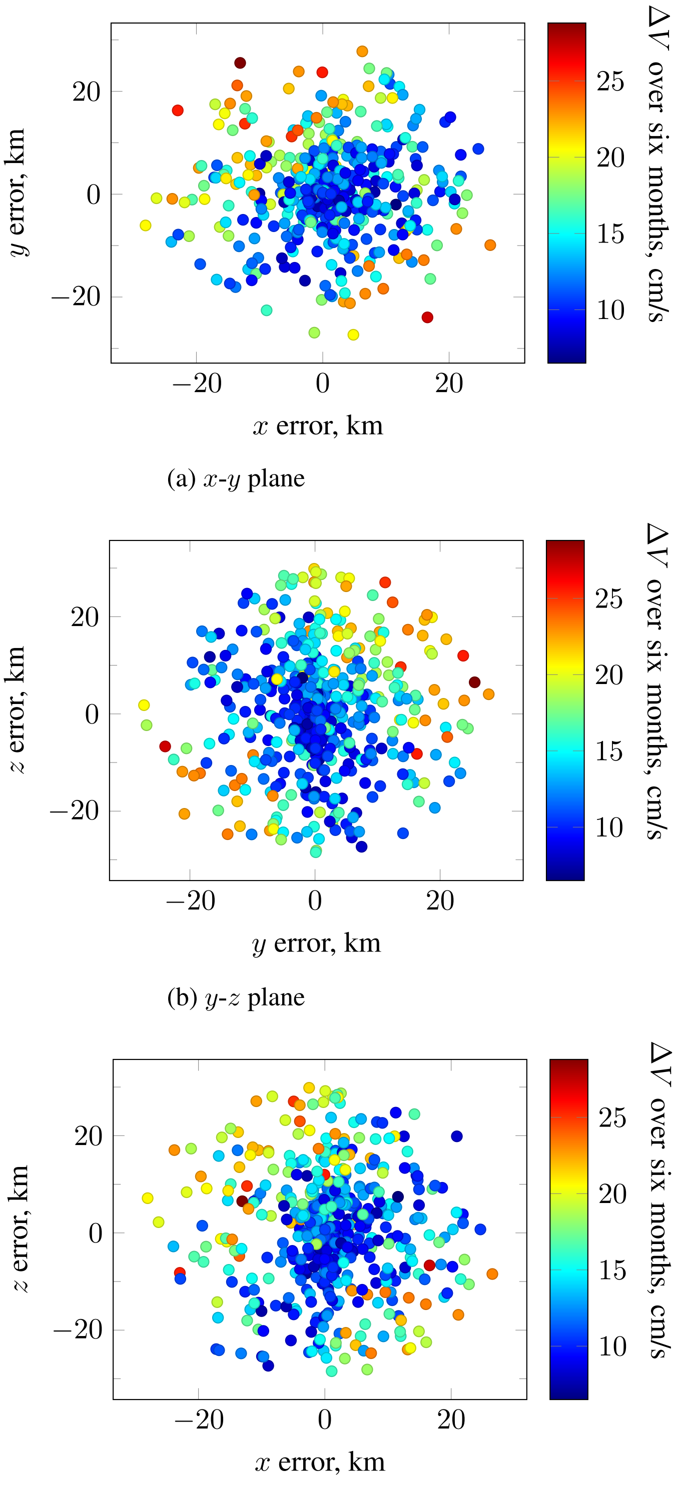


Figure 6: Histogram of cumulative ΔV over 56 revolutions from 500 Monte-Carlo samples with xz-plane crossing control (standard deviation $= 22.04$ cm/s)

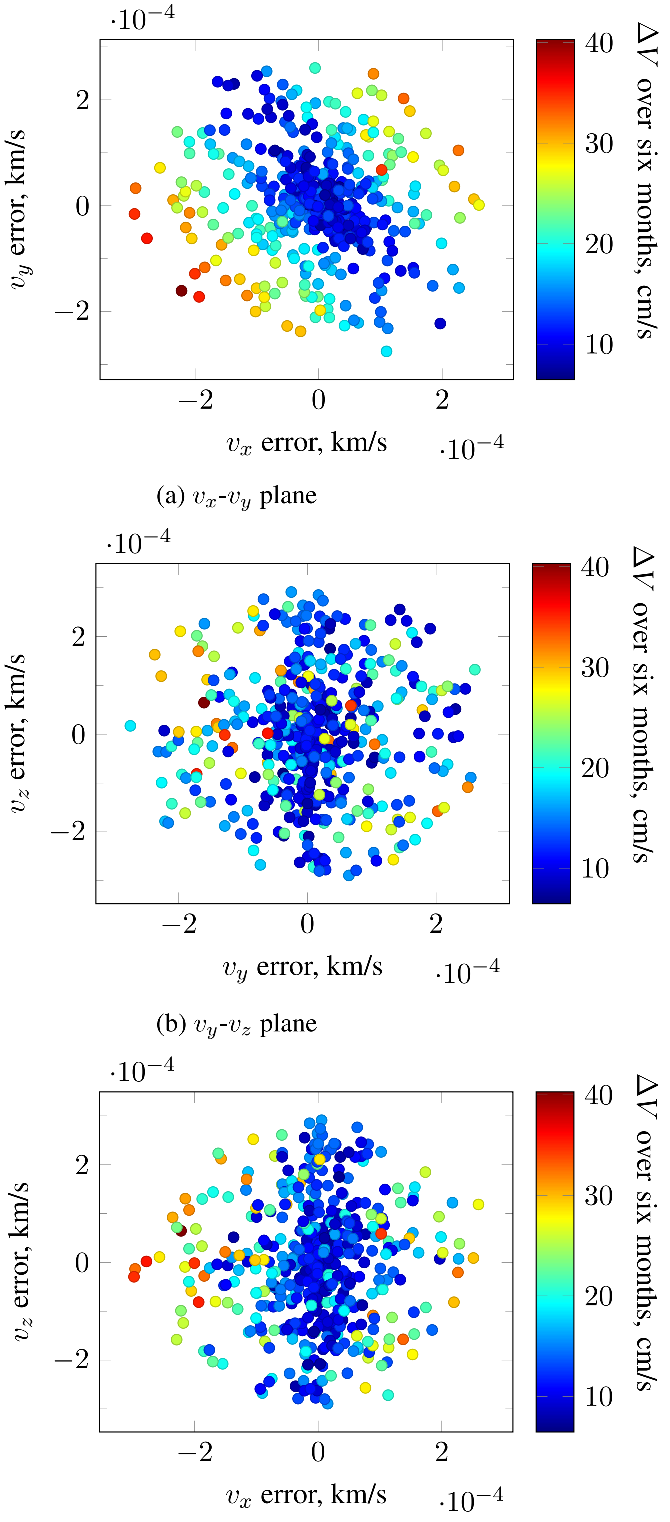


(a) x-y plane

(b) y-z plane

(c) x-z plane

Figure 7: Cumulative station-keeping ΔV over 28 revolutions with zero velocity insertion error and 3-$\sigma_{r_i} = 30$ km random position insertion errors. Error components are resolved in the Earth-Moon rotating frame.



(a) v_x-v_y plane

(b) v_y-v_z plane

(c) v_x-v_z plane

Figure 8: Cumulative station-keeping ΔV over 28 revolutions with zero position insertion error and 3-$\sigma_{v_i} = 30$ cm/s random velocity insertion errors. Error components are resolved in the Earth-Moon rotating frame.

in terms of position, velocity, and epoch. This network was trained and tested under orbital insertion errors and navigational uncertainties. When factoring out the transient cost associated with stabilizing the insertion error, the DRL policy provided a mean steady-state gradient of 22.8 m/s per 56 NRHO revolutions. The DRL performance was compared to xz-plane crossing control, a well-known heuristic control scheme, and bested it in both consistency (standard deviation of the annual station-keep cost) and magnitude. Future work may consider the use of the DRL framework together with a navigation filter to analyze the DRL's effectiveness in a scenario that is closer to actual station-keeping operation.

REFERENCES

[1] D. Lee, "Gateway Destination Orbit Model: A Continuous 15 Year NRHO Reference Trajectory," tech. rep., NASA Johnson Space Center, 2019.

[2] M. Duggan, X. Simon, and T. Moseman, "Lander and cislunar gateway architecture concepts for lunar exploration," *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–9.

[3] T. A. Pavlak, *Trajectory design and orbit maintenance strategies in multi-body dynamical regimes*. PhD thesis, Purdue University, 2013.

[4] J. Williams, D. E. Lee, R. J. Whitley, K. A. Bokelmann, D. C. Davis, and C. F. Berry, "Targeting cislunar near rectilinear halo orbits for human space exploration," *AAS/AIAA Space Flight Mechanics Meeting*, No. JSC-CN-38615, 2017.

[5] K. C. Howell and T. M. Keeter, "Station-keeping Strategies for Libration Point Orbits: Target Point and Floquet Mode Approaches," *Spaceflight Mechanics*, 1995, pp. 1377–1396.

[6] G. Gómez, K. Howell, J. Masdemont, and C. Simó, "Station-keeping strategies for translunar libration point orbits," *Advances in Astronautical Sciences*, Vol. 99, No. 2, 1998, pp. 949–967.

[7] P. Elango, S. Di Cairano, U. Kalabić, and A. Weiss, "Local Eigenmotion Control for Near Rectilinear Halo Orbits," *2022 American Control Conference*, IEEE, 2022, pp. 1822–1827.

[8] D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Stationkeeping analysis for spacecraft in lunar near rectilinear halo orbits," *27th AAS/AIAA Space Flight Mechanics Meeting*, Vol. 160, American Astronautical Society San Antonio, Texas, 2017, pp. 3199–3218.

[9] E. M. Zimovan, K. C. Howell, and D. C. Davis, "Near rectilinear halo orbits and their application in cis-lunar space," *3rd IAA Conference on Dynamics and Control of Space Systems, Moscow, Russia*, Vol. 20, 2017, p. 40.

[10] L. Bucci, M. Lavagna, R. Jehn, *et al.*, "Station keeping techniques for near rectilinear orbits in the Earth–Moon system," *Proceedings of 10th international ESA conference on GNC systems, Salzburg, Austria*, Vol. 29, 2017.

[11] D. Davis, S. Bhatt, K. Howell, J.-W. Jang, R. Whitley, F. Clark, D. Guzzetti, E. Zimovan, and G. Barton, "Orbit maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits," *AAS/AIAA Space Flight Mechanics Meeting*, No. JSC-CN-38626, 2017.

[12] V. Muralidharan and K. C. Howell, "Stationkeeping in Earth-Moon Near Rectilinear Halo Orbits," *AAS/AIAA Astrodynamics Specialist Conference, South Lake Tahoe, California, USA*, 2020.

[13] C. P. Newman, D. C. Davis, R. J. Whitley, J. R. Guinn, and M. S. Ryne, "Stationkeeping, orbit determination, and attitude control for spacecraft in near rectilinear halo orbits," 2018.

[14] D. C. Davis, S. M. Phillips, K. C. Howell, S. Vutukuri, and B. P. McCarthy, "Stationkeeping and transfer trajectory design for spacecraft in cislunar space," *AAS/AIAA Astrodynamics Specialist Conference*, Vol. 8, 2017.

[15] P. Elango, S. Di Cairano, K. Berntorp, and A. Weiss, "Sequential linearization-based station keeping with optical navigation for NRHO,"

[16] N. B. LaFarge, K. C. Howell, and D. C. Folta, "An Autonomous Stationkeeping Strategy for Multi-Body Orbits Leveraging Reinforcement Learning," *Conference: AIAA SciTech ForumAt: San Diego, CA*, American Institute of Aeronautics and Astronautics, 2022.

[17] S. Bonasera, I. Elliott, C. J. Sullivan, N. Bosanac, N. Ahmed, and J. McMahon, "Designing Impulsive Station-Keeping Maneuvers Near a Sun-Earth Halo Orbit via Reinforcement Learning," 2021.

[18] D. Guzzetti, "Reinforcement Learning and Topology of orbit Manifolds for Stationkeeping of Unstable Symmetric Periodic Orbits," 2019.

[19] S. Bonasera, N. Bosanac, C. J. Sullivan, I. Elliott, N. Ahmed, and J. W. McMahon, "Designing Sun-Earth L2 Halo Orbit Stationkeeping Maneuvers via Reinforcement Learning," 2022.

[20] A. Scorsoglio, R. Furfaro, R. Linares, and M. Massari, "Relative motion guidance for near rectilinear lunar orbits with path constraints via actor-critic reinforcement learning," 2023.

[21] A. Scorsoglio, R. Furfaro, R. Linares, and M. Massari, "Actor-Critic Reinforcement learning Approach to Relative Motion Guidance in Near-Rectilinear Orbit," 2019.

[22] N. B. LaFarge, K. C. Howell, and D. C. Folta, "Adaptive Closed-Loop Maneuver Planning for Low-Thrust Spacecraft using Reinforcement Learning," *Acta Astronautica*, Vol. 211, 10 2023, pp. 142–154, 10.1016/j.actaastro.2023.06.004.

[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 1 2018.

[24] C. H. Acton, "Ancillary data services of NASA's Navigation and Ancillary Information Facility," *Planetary and Space Science*, Vol. 44, No. 1, 1996, pp. 65–70. Planetary data system, https://doi.org/10.1016/0032-0633(95)00107-7.

[25] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*. Fundamentals of Astrodynamics and Applications, Microcosm Press, 2001.

[26] J. Williams, D. E. Lee, R. J. Whitley, K. A. Bokelmann, D. C. Davis, and C. F. Berry, "Targeting Cislunar Near Rectilinear Halo Orbits for Human Space Exploration," 2017.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[28] T. Suda and D. Nikovski, "Deep Reinforcement Learning for Optimal Sailing Upwind," *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8.

[29] D. C. Davis, S. A. Bhatt, K. C. Howell, J. W. Jang, R. J. Whitley, F. D. Clark, D. Guzzetti, E. M. Zimovan, and G. H. Barton, "Orbit maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits," Vol. 160, 2017, pp. 2257–2276.

[30] B. Gough, *GNU scientific library reference manual*. Network Theory Ltd., 2009.