AAS 19-680

REINFORCEMENT LEARNING AND TOPOLOGY OF ORBIT MANIFOLDS FOR STATIONKEEPING OF UNSTABLE SYMMETRIC PERIODIC ORBITS

Davide Guzzetti*

This work investigates reinforcement learning (RL) as an algorithm for orbit stationkeeping within chaotic environments. We first consider maintenance of unstable symmetric periodic (USP) orbits within circular restricted three-body problem (CR3BP) dynamics. Because topology for USP orbit dynamics is largely understood, USP orbits may be a testing ground to explore maintenance strategies based on RL models. Existing stationkeeping algorithms, including Floquet mode and gradient-based optimal control, may also supply a reference for characterizing RL performance. Outlining fundamental RL mechanisms for orbit stationkeeping and describing their relation to existing orbit maintenance techniques will support similar applications within more complex scenarios.

INTRODUCTION

NASA, with the support of other international space agencies, is preparing for a new era of exploration of the solar system.[1] In the next decades, the number of robotic and crew-tended vehicles that will operate within gravitational multi-body fields is expected to increase. Gravitational multi-body dynamics impact the motion of a large array of space applications, including astrophysical observatories,[2] manned habitats orbiting in cislunar space,[3] probes searching for extraterrestrial life,[4] and miniaturized satellites operating at small bodies *.

One key challenge in flying spacecraft within multi-body environments is the efficient and effective maintenance of the vehicle along a desired path. Spacecraft flight within multi-body environments requires significant analytical effort and extensive domain knowledge[5,6] to determine optimal maneuver location, size and direction for stationkeeping. Consequently, adaptive and autonomous guidance solutions are warranted for the next decade of NASA-planned missions to reduce cost and complexity of long-term operations. In addition to cost reduction, autonomous maneuver planning may enable new mission classes by alleviating navigation requirements and introducing near-term re-planning capabilities. For example, remote regions of the solar system, where operation of traditional spacecraft may be subject to round-trip light time constraints, may become more accessible with autonomous navigation and maneuver planning capabilities. These objectives are well aligned with NASA technology roadmaps.[7]

There exist different orbit maintenance algorithms for multi-body dynamics, including methods adapted from classical control schemes,[8,9] methods based on dynamical systems theory,[10–13] optimal targeting of way points along a fixed baseline solution[11,13,14] and enforcing the direction of the velocity vector when an Earth-Moon-Spacecraft alignment occurs.[6,15] These methods have supported several missions[16] in the past, but have not yielded fully autonomous maneuver planning capabilities. A currently common practice is to combine existing trajectory control methods with system identification techniques to develop indirect forms of adaptive control: given an approximate system model, first identify system parameters from data collected during flight and, then, plan maneuvers based on the reconstructed model. In this process, maneuver planning

*Assistant Professor, Department of Aerospace Engineering, 328 Davis Hall, Auburn University, Auburn, Alabama, 36849, guzzetti@auburn.edu

*https://www.nasa.gov/feature/small-satellite-concept-finalists-target-moon-mars-and-beyond/, visited on July 2019

remains a challenging task, as it demands a considerable effort for constructing solutions that are, then, only applicable to very specific vehicles operating in narrow dynamical regimes. As today, maneuver planning within multi-body environment is executed under the supervision of human operators at the ground station. Current orbit maintenance algorithms are difficult to automatize because they rely on the assumption that spacecraft dynamics are, eventually, well-modeled and that baseline trajectories are known and given as an input. Establishing a sustainable human presence, robotic or actual, in the solar system will require algorithms that are not only able to calculate feasible maneuvers, but that can also autonomously adapt to both short and long-term variations of the mission scenario. The ideal autonomous spacecraft is one that is able to follow a desired trajectory regardless of the dynamical environment that it experiences. Accomplishing that goal will require astrodynamicists to develop more robust and reliable frameworks to incorporate existing maneuver planning algorithms into an autonomous guidance scheme with online learning and adaptation capabilities.

Reinforcement learning (RL) may supply a framework to design orbit maintenance algorithms that are as accurate as current algorithms, but also more adaptable and robust to uncertainties. Reinforcement learning is a data-oriented approach that can solve general path-planning problems by cleverly sampling numerous state-action pairs and promoting the combinations of state and action that are more likely to produce a desired outcome.[17] Reinforcement learning algorithms do not require, in principle, an analytical description of the system dynamics to function, because they determine optimal control actions solely from the observation of state transitions (i.e., the process of changing from the current state to another). Certain reinforcement learning algorithms are capable of both offline and online learning;[17] such algorithms may be pre-trained offline at the ground station within simulated environments (to meet optimality conditions) and continue learning online to adapt to uncertainties. Given its characteristics, reinforcement learning has become an effective toolkit within robotic applications to reproduce hard-to-engineer behavior and design control solution that are less reliant on knowledge of system dynamics.[18,19] Reinforcement learning recent successes in ground robotics poses the question whether it is applicable to spacecraft maneuver planning within gravitational multi-body dynamics.

To become effective, reinforcement learning algorithms need to visit each significant state-action pair a sufficient number of times during the sampling process. That can be difficult to accomplish within continuous state-action systems, including orbit maintenance dynamics, where it is not possible to visit all possible state-action pairs in a finite time duration. Then, it is necessary to map continuous state-action spaces to solution spaces with tractable dimension via discretization and/or parametrization.[17,20–22] The map from a continuous to a discrete or parametric solution space is critical for determining the success of reinforcement learning algorithms, especially within multi-body dynamics. Multi-body dynamics may exhibit chaotic behavior that complicates the discretization/parametrization process of the state-action space. In fact, if dynamics are chaotic, it is possible for infinitesimally close states, or actions, to yield strikingly different solutions. Parameters such as the probability of random over optimal actions, the number of trials to identify an optimal action for each state, and the value of each action toward optimality also shape the behavior of reinforcement learning algorithms. An understanding of the impact of reinforcement learning parameters and solution space representations on orbit maintenance of multi-body motion is warranted.

In this work, we explore fundamental mechanisms behind reinforcement learning as an algorithm for orbit stationkeeping within chaotic environments. We start from considering the orbit maintenance of unstable symmetric periodic (USP) orbits within circular restricted three-body problem (CR3BP) dynamics. Because topology of USP orbits and dynamics in their neighborhood are largely understood, USP orbits may serve as a testing ground for the application of reinforcement learning to orbit maintenance. Existing stationkeeping algorithms, including Floquet mode and gradient-based optimal control, may also supply a reference for characterizing reinforcement learning performance. Floquet mode controllers reflect the application of analytical effort and domain knowledge to orbit stationkeeping.[23] Gradient-based optimization may provide a benchmark for optimal control solutions. Outlining reinforcement learning mechanisms for orbit stationkeeping within a well-known orbit environment, and describing their relation to existing orbit maintenance techniques will support the application of reinforcement learning algorithms to more complex space mission scenarios.

BACKGROUND INFORMATION

Dynamical Model

The investigation of reinforcement learning as a framework for stationkeeping of multi-body orbits is first based on CR3BP modeling of spacecraft dynamics. The CR3BP model renders an approximated description of orbit dynamics for a massless spacecraft under the attraction of two point-mass bodies that move in circular orbits about their mutual barycenter. The motion of the vehicle is described relative to a coordinate frame, $\hat{x}\hat{y}\hat{z}$, that rotates with the circular motion of the attracting bodies. In such rotating frame, the spacecraft is located by nondimensional coordinates (x, y, z) and its orbit equations of motion are written as:

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x},\ \ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y},\ \ddot{z} = \frac{\partial U}{\partial z} \tag{1}$$

where the pseudo-potential function,

$$U = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{d} + \frac{\mu}{r}$$

while

$$d = \sqrt{(x+\mu)^2 + y^2 + z^2}$$

and

$$r = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$$

The mass parameter, μ, denotes the ratio of the smaller attracting mass over the total mass of the system. The dynamical model in Eq. (1) serves to define state transitions within a simulated environment where the reinforcement learning agent trains. Although the dynamical environment is virtually known, the reinforcement learning agent does access such information to determine optimal actions for orbit maintenance.

Unstable Symmetric Periodic Orbits

CR3BP equations of motion admit five relative equilibrium points and infinitely many periodic solutions.[24] Expressing the general solution of the equations of motion as dynamical flow, $\varphi(\boldsymbol{x}, t)$, from an initial state, $\boldsymbol{x}$, then, any state, $\boldsymbol{x}_0$, that belongs to a periodic orbit, Γ, satisfies

$$\boldsymbol{x}_0 = \varphi(\boldsymbol{x_0}, P) \quad \text{for} \quad \boldsymbol{x}_0 \in \Gamma \tag{2}$$

where P is the minimum time span between the recurrence of state $\boldsymbol{x}_0$, or simply the orbit period. Solutions to Eq. (2) depends on the vector basis that describes state $\boldsymbol{x}_0$. In CR3BP applications the vector basis is typically, but not necessarily, defined by the rotating frame, such that orbit periodicity is determined relative to an observed fixed in that frame. We will adopt this convention, unless stated otherwise.

Within CR3BP dynamics, spectral analysis may be employed to determine linear stability features of a reference periodic orbit. Along any reference periodic or aperiodic trajectory, the state transition matrix (STM), $\Phi(t, 0)$, describes the linear mapping between a variation of the initial state, $\delta\boldsymbol{x}_0$, and the consequent variation of the final state, $\delta\boldsymbol{x}(t)$, such that $\delta\boldsymbol{x}(t) = \Phi(t, 0)\delta\boldsymbol{x}_0$. The STM is a solution to the following set of differential equations

$$\begin{cases} \dot{\Phi}(t, 0) = J(t)\Phi(t, 0) \\ \Phi(0, 0) = \mathbb{I} \end{cases} \tag{3}$$

where $J(t)$ is the Jacobian matrix for the CR3BP equations of motion. When the STM is evaluated along a periodic orbit and for an orbit period, it is called monodromy matrix $M = \Phi(P, 0)$. Properties of the monodromy matrix are fundamental to define the linear stability structure of the reference periodic orbit.

Linear stability features of a periodic orbit are derived from the eigenvalues and eigenvectors of the monodromy matrix. Eigenvalues for the monodromy matrix λ_i, real or complex, with modulus value lower than one indicate the existence of stable modes. The corresponding eigenvectors define a linear subspace that locally renders the stable manifold of the reference orbit. Eigenvalues λ_i such that $|\lambda_i| > 1$ are associated with

unstable modes. The corresponding eigenvectors define a linear subspace that locally renders the unstable manifold of the reference orbit. Finally, marginally linear stable modes correspond to $|\lambda_i| = 1$. A periodic reference orbit is unstable if any of the eigenvalues possesses a modulus greater than one, i.e. $\exists\ |\lambda_i| > 1$; it is marginally stable otherwise (in the linear approximation). The monodromy matrix for periodic orbits within CR3BP dynamics has a peculiar spectral structure. For any periodic orbit, eigenvalues always appear in reciprocal pairs: such that, if there exist λ_i, then there must exist $1/\lambda_i$. As a consequence, the existence of stable manifolds for a reference periodic orbit is always associated with the existence of unstable manifolds, and vice-versa. Stationkeeping becomes mandatory to maintain a spacecraft along unstable periodic orbits.

In this work, we explore stationkeeping of unstable periodic orbits that are symmetric about the xz-plane of the rotating frame. Symmetry of the gravitational potential in the $\hat{y}$ direction within the equations of motion is often leveraged to identify periodic motion, making symmetric orbits a very common solution. Upon symmetry principles, multiple algorithms are available to numerically compute symmetric periodic orbits.[25] Lyapunov, distant retrograde, halo (including near rectilinear halo) orbits and several known families of resonant trajectories are, in fact, symmetric orbits[26] Targeting symmetric conditions at half orbit revolution is also an effective form of orbit maintenance strategy within higher-fidelity dynamics. In fact, retention of nearly symmetric geometry within higher-fidelity ephemeris models may be possible via small ΔV maneuvers. To date, the vast majority of satellites that have flown along multi-body orbit displays orbit motion that is traceable to one of the CR3BP symmetric periodic orbit.[8]

Unstable symmetric periodic orbits are a class of CR3BP trajectories that is sufficiently general, deeply understood, and largely utilized in real mission applications. As such, USP orbits are an ideal testing ground to demonstrate the application of reinforcement learning within multi-body dynamics.

Stationkeeping

When the reference orbit is unstable, the spacecraft necessarily tends to depart from the reference motion as consequence of small perturbations. The objective of a stationkeeping algorithm is to determine maneuvers that offset the orbit departure for a given number of revs. Computing stationkeeping maneuvers typically requires to monitor a trajectory error vector, $\boldsymbol{\delta}$, which is defined as the isochronous difference in position and velocity between the current and reference trajectory. In this work, three approaches are considered to determine orbit correction maneuvers: Floquet mode, gradient-based optimization, and reinforcement learning.

Floquet mode. Floquet mode algorithms determine stationkeeping maneuvers so to cancel the projection of the trajectory error vector, $\boldsymbol{\delta}$, in the direction orthogonal to the stable manifold.[11,13] The direction of the unstable manifold at the maneuver location is approximated by the linear unstable subspace describing the local neighborhood of the reference orbit. The correction maneuver vector $\boldsymbol{\Delta V} = [\Delta V_x, \Delta V_y, \Delta V_z]^T$ is computed by solving the equation

$$\Delta V_x \Pi_4 + \Delta V_y \Pi_5 + \Delta V_z \Pi_6 + \alpha = 0 \tag{4}$$

where the vector $\boldsymbol{\Pi} = [\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6]^T$ is such that

$$E^T \boldsymbol{\Pi} = [1, 0, 0, 0, 0, 0]^T$$

and $\alpha = \boldsymbol{\delta}^T \boldsymbol{\Pi}$. The matrix E contains ordered eigenvectors of the monodromy matrix for the reference periodic orbit; the first column of matrix E renders the given unstable subspace, assuming one-dimensional unstable manifold.

Gradient-based optimization. Stationkeeping maneuvers are determined to minimize a user-defined objective function $C(\boldsymbol{\xi})$ while the solution is subject to a set of constraints $\boldsymbol{g}(\boldsymbol{\xi}) \leq \mathbf{0}$. The vector $\boldsymbol{\xi}$ describes the free variable vector for optimization of the cost function, and includes the orbit maintenance maneuvers. Depending on the formulation of the stationkeeping problem, the total ΔV and the trajectory error vector, $\boldsymbol{\delta}$,

may be incorporated into the objective function or the constraint vector function. The optimal solution $\boldsymbol{\xi}^*$ is achieved by finding the minimum of a cost function C

$$\begin{aligned} &\min C(\boldsymbol{\xi}) \\ &\text{subject to} \quad \boldsymbol{g}(\boldsymbol{\xi}) \leq \mathbf{0} \end{aligned} \tag{5}$$

Techniques that may be employed to solve the minimization problem in Eq. (5) include sequential quadratic programming.

Reinforcement learning. Stationkeeping maneuvers are learned from interaction of a control algorithm, called an agent, with the dynamical environment to achieve orbit maintenance. First define a series of waypoints along the trajectory. At each waypoint the agent receives a representation of the spacecraft position and velocity $\boldsymbol{s}_i \in S$, where S is a finite set of all possible states. On the basis of the state $\boldsymbol{s}_i \in S$ the agent selects a control maneuver $\boldsymbol{\Delta V}_i \in A$, where A is a finite set of all available actions. At the next waypoint, as a consequence of the control maneuver $\boldsymbol{\Delta V}_i$, the agent receives a scalar reward r_i and a new state $\boldsymbol{s}_{i+1}$. The agent's goal is to maximize the future accumulated, discounted reward, $G_t = \sum_{(i=t)}^{\infty} \gamma^{(i-t)} r_i$, where $\gamma \leq 1$ is a discount factor. Most reinforcement learning algorithms are based on estimating future accumulated rewards in the form of a state-action value function

$$Q^{\pi}(\boldsymbol{s}, \boldsymbol{\Delta V}) = E_{\pi}\{G_t | \boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{\Delta V}_t = \boldsymbol{\Delta V}\} \tag{6}$$

which describes the expected value of a state-action pair under a certain control policy, $\boldsymbol{\Delta V} = \pi(\boldsymbol{s})$. After $Q^{\pi}(\boldsymbol{s}, \boldsymbol{\Delta V})$ is available, control maneuvers can be determined at each state $\boldsymbol{s}_i$ as

$$\boldsymbol{\Delta V}_i = \arg\max_{\boldsymbol{\Delta V}} Q^{\pi}(\boldsymbol{s}_i, \boldsymbol{\Delta V}) \tag{7}$$

The method employed to estimate the state-action value function in Eq. (6) and the corresponding policy determines the class of RL algorithm. If actions are computed directly from Eq. (7), then the RL algorithm falls into the category of action-value methods. Action-value methods are particularly attractive for their simplicity, as they may be conveniently implemented as tabular methods. Action-value methods are better suited for discrete and low-dimensional action spaces. Solving for Eq. (7) in continuous spaces may quickly become computationally intensive by requiring global maximization at each iteration of the learning process. Policy gradient methods avoid solving for Eq. (7) by directly updating the policy in the direction of the gradient of the state-action value function, Q.[17] In virtue of their simplicity and intuitive interpretation, our work first investigates tabular action-value methods for the stationkeeping of USP orbits within multi-body dynamics.

REINFORCEMENT LEARNING FOR USP ORBIT MAINTENANCE

Formulation of the Targeting Problem

An effective strategy to stationkeep USP orbits is to control crossing conditions at the x-axis. Maintenance maneuvers are applied every half orbit revolution at the trajectory intersection with the x-axis to target predetermined conditions at the next crossing. This form of multi-body orbit stationkeeping is labeled crossing control. Crossing control may be formulated as a reinforcement learning problem following algorithm 1, where rewards are designed so that the optimal policy may achieve orbit maintenance. The resulting RL formulation for the targeting problem is visually described in Figure 1. Looping through algorithm 1 generates a sequence

$$\mathcal{E} : \{\boldsymbol{s}_1, \boldsymbol{\Delta V}_1, r_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_i, \boldsymbol{\Delta V}_i, r_i, \boldsymbol{s}_{i+1}, \ldots, \boldsymbol{s}_T\}$$

that defines a training episode. A training episode essentially corresponds to a single orbit maintenance simulation. Transition from the current state, $\boldsymbol{s}_i$, to the next, $\boldsymbol{s}_{i+1}$ occurs through the propagation of orbit dynamics. In this work we consider both linear transitions, $\boldsymbol{s}_{i+1} = A\boldsymbol{s}_i + B\boldsymbol{\Delta V}_i$, which approximate orbit dynamics near a reference orbit, and nonlinear transitions $\boldsymbol{s}_{i+1} = \boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i)$, which render the fully

Algorithm 1 Reinforcement learning pseudo-code for USP orbit maintenance

while $\boldsymbol{s}_i$ is not terminal **do**
  Estimate spacecraft state, $\boldsymbol{s}_i$, at x-axis crossing
  Compute action $\boldsymbol{\Delta V}_i$ based on current policy
  Apply action $\boldsymbol{\Delta V}_i$
  Transition to the next state by propagating orbit dynamics to next crossing (may include stochastic orbit errors)
  Get reward r_i
  Get new state, $\boldsymbol{s}_{i+1}$
end while

nonlinear Eq. (1). We also explore training of RL agents on both deterministic transitions, $\mathbb{P}(\boldsymbol{s}_{i+1} = \boldsymbol{s}' | \boldsymbol{s}_i = \boldsymbol{s}, \boldsymbol{\Delta V}_i = \boldsymbol{\Delta V}) = 1$, and stochastic transitions, $\mathbb{P}(\boldsymbol{s}_{i+1} = \boldsymbol{s}' | \boldsymbol{s}_i = \boldsymbol{s}, \boldsymbol{\Delta V}_i = \boldsymbol{\Delta V}) \leq 1$, that may include random orbit determination errors.

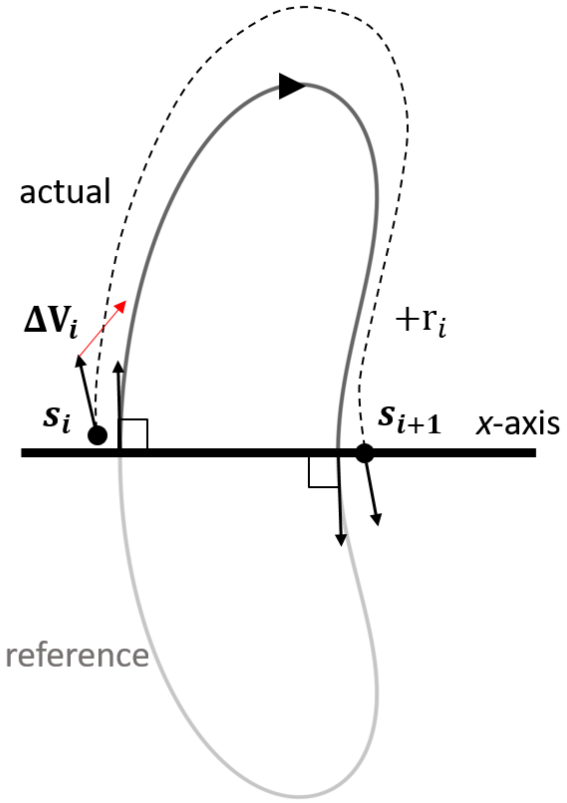


Figure 1: Schematics of the targeting problem.

Monte Carlo Control

This section describes a possible implementation and training of RL agents for USP orbit maintenance using a Monte Carlo control approach. Assume that the RL agent adopts an ϵ-greedy policy, so that correction

maneuvers during training are determined as

$$\boldsymbol{\Delta V}_i = \begin{cases} \arg\max_{\boldsymbol{\Delta V}} Q(\boldsymbol{s}_i, \boldsymbol{\Delta V}) & \text{with probability } 1-\epsilon \\ \text{random} & \text{with probability } \epsilon \end{cases} \tag{8}$$

Since the value of each state-action pair is unknown before training, the objective of the learning process is to reconstruct the state-action value function, Q. The random actions that are included in the ϵ-greedy policy allow the agent to extend the exploration of the solution space during learning. The epsilon value, ϵ, should decrease inversely proportional to the number of training episodes, k, to ensure convergence on an optimal state-action value function.[17] Monte Carlo agents are trained offline because their training requires access to the complete episode sequence and the final episode return, G_k. Sample the k-th episode using the current policy π_k, $\mathcal{E}_k : \{\boldsymbol{s}_1, \boldsymbol{\Delta V}_1, r_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_i, \boldsymbol{\Delta V}_i, r_i, \boldsymbol{s}_{i+1}, \ldots, \boldsymbol{s}_T\}$. For each state and action in the sequence, keep a total count of state-action occurrences over all episodes

$$N(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i) \leftarrow N(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i) + 1 \tag{9}$$

and update the state-action value function using the current episode return, G_k

$$Q(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i) \leftarrow Q(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i) + \alpha_k (G_k - Q(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i)) \tag{10}$$

where α_k defines the learning rate and is a function of the number of visits to the current state-action pair

$$\alpha_k = \frac{1}{N(\boldsymbol{s}_i, \boldsymbol{\Delta V}_i)} \tag{11}$$

A schedule is also defined for the epsilon value to ensure convergence on the optimal policy, such that

$$\epsilon(\boldsymbol{s}_i) = \frac{N_0}{N(\boldsymbol{s}_i) + N_0} \tag{12}$$

where the parameter N_0 is introduced to delay the contraction of the epsilon value and enable more random actions during the early phases of the learning process to explore the solution space.

Construction of State Features

Spacecraft orbit state may be described within a real, continuous coordinate space of six dimensions, $\mathbb{R}^6$. For stationkeeping purposes, it may be convenient to represent the spacecraft state as the trajectory error at the crossing of the x-axis relative to a reference trajectory

$$\boldsymbol{\delta} = \boldsymbol{x} - \boldsymbol{x}_{ref} = [\delta_x, \delta_y, \delta_z, \delta_{\dot{x}}, \delta_{\dot{y}}, \delta_{\dot{z}}]^T$$

where $\boldsymbol{x}$ are the actual spacecraft position and velocity coordinates at the epoch of crossing, and $\boldsymbol{x}_{ref}$ describes the crossing conditions along the reference orbit. For USP orbits, the crossing conditions may reflect symmetry properties of the trajectory geometry. Given the trajectory error vector, $\boldsymbol{\delta}$, the RL agent state may be described by a finite and discrete set of table lookup features, $\boldsymbol{s}_\delta \in S \subset \mathbb{Z}^6$, that are constructed via state aggregation. Each element of the state feature vector, $\boldsymbol{s}_\delta$, corresponds to an element of the trajectory error vector, δ, such that

$$s_\delta = \begin{cases} \left\lfloor \dfrac{\delta}{\Delta s} \right\rfloor & \text{if} \quad 0 \leq \delta \leq \delta_{\max} \\ \left\lceil \dfrac{\delta}{\Delta s} \right\rceil & \text{if} \quad \delta_{\min} \leq \delta < 0 \end{cases} \tag{13}$$

where Δs denotes the size of the discretization intervals for each element. When the vector component δ is larger then $\delta_{\max}$, then $s_\delta = \left\lfloor \dfrac{\delta_{\max}}{\Delta s} \right\rfloor$. Similarly, $s_\delta = \left\lceil \dfrac{\delta_{\min}}{\Delta s} \right\rceil$ for $\delta < \delta_{\min}$. Equation (13) is a zero order polynomial representation of the error vector components. Along a periodic orbit the Markov chain of states

may also become periodic. To break the periodicity of the Markov chain, it may be useful to augment the RL agent state feature vector $\boldsymbol{s} = [s_{\text{crossing}}, \boldsymbol{s}_\delta^T]^T$ with a feature $s_{\text{crossing}} \in \mathbb{Z}$ that describes the cumulative number of x-axis crossings. The dimension of the RL agent state feature space grows rapidly as the discretization of the continuous spacecraft state is refined. In fact, the number of possible discrete states that describes the RL agent state is proportional to n^6 (excluding the feature s_{crossing}), where n is the number of aggregation intervals that is employed for each state feature.

Orbit manifolds may provide an attractive dynamical structure to reduce the dimensionality of the RL agent state feature space. Similar to the idea behind Floquet mode control, if we assume linear (or linearized) dynamics - the superposition principle applies. According to the superposition principle, only the components of the error along the unstable manifold are responsible for spacecraft departure from the reference trajectory. Behavior that resembles the effects of the superposition principle may also be visible within nonlinear dynamics. Along certain USP orbits, an initial randomly distributed error may tend to align along the unstable manifold direction as it is propagated within fully nonlinear dynamics (see Figure 2). Following

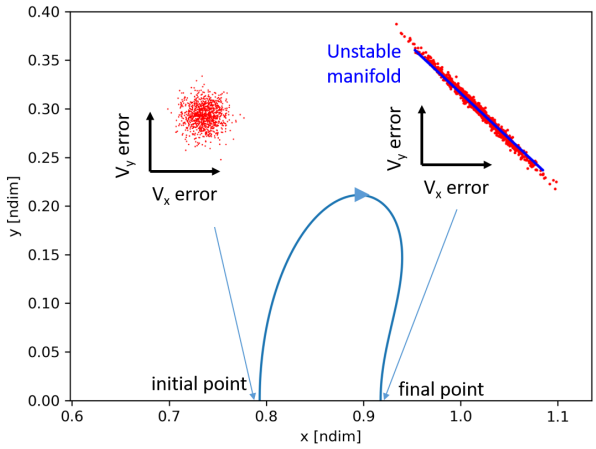


Figure 2: Propagation of uniformly distributed orbit states nearby the reference trajectory.

the observation that the unstable manifold dominates the spacecraft orbit dynamics nearby the reference, the full representation of spacecraft state may be replaced with its projection on the unstable manifold linear subspace. Then, alternative RL state features may be defined as $\boldsymbol{s} = [s_{\text{crossing}}, \boldsymbol{s}_{W^u}]^T$ with

$$s_{W^u} = \begin{cases} \left\lfloor \dfrac{\delta_{W^u}}{\Delta s} \right\rfloor & \text{if} \quad 0 \leq \delta \leq \delta_{\max} \\ \left\lceil \dfrac{\delta_{W^u}}{\Delta s} \right\rceil & \text{if} \quad \delta_{\min} \leq \delta < 0 \end{cases} \tag{14}$$

where δ_{W^u} indicates the elements of the trajectory error projection onto the unstable manifold linear subspace

$$\boldsymbol{\delta}_{W^u} = W^u \boldsymbol{\delta}$$

and W^u is a projection matrix. The dimension of the projection $\boldsymbol{\delta}_{W^u}$ is one for planar dynamics, one or two for spatial dynamics (depending on the dimension of the unstable manifold subspace). Therefore, the number of possible discrete states that describes the RL agent state may be reduced up to n (excluding feature s_{crossing}) using Eq. (14). The selection of state features may significantly impact the performance and effectiveness of RL algorithms for USP orbit stationkeeping.

Reward Definition

Within classical stationkeeping algorithms, orbit correction maneuvers are computed to target specific conditions or optimize a given objective function. An RL agent optimal policy is, instead, driven by rewards. A reward function replaces the role that targeting conditions and/or objective functions have within classical orbit maintenance schemes. A reward function for the maintenance of a reference USP orbit may be defined as follows

$$r_i = -w_{\Delta V}||\boldsymbol{\Delta V}_i|| + w_{rev}i - w_e e_i \tag{15}$$

where i denotes the current step in the Markov chain and coincides with the current count of x-axis crossings; $w_{\Delta V}$, w_{rev}, and w_e are scalar weights; e_i is a measure of the trajectory error. Essentially, a function as that in Eq. (15) rewards the RL agent for surviving longer and penalizes it for committing larger trajectory errors or requiring larger velocity corrections. An episode may be terminated when a certain number of crossing count, i, is reached or when the trajectory error measure, e_i, exceeds a certain threshold. Possible measures of the trajectory error includes the norm of the error

$$e_i = ||\boldsymbol{\delta}_i||, \tag{16}$$

the error on the j-th spacecraft state components

$$e_i = |\delta_{j,i}|, \tag{17}$$

and a linear estimate of the error norm after p orbit revolutions

$$e_i = ||M^p \boldsymbol{\delta}_i|| \tag{18}$$

recalling that M denotes the monodromy matrix. An error measure as in Eq. (17) may be useful to target specific conditions at the x-axis, such as perpendicular crossing. An error measure as in Eq. (18) may be employed to capture the asymptotic behavior of the current trajectory.

RL agents may behave unexpectedly during the learning process, especially when compared to traditional targeting and optimization algorithms. Undesirable behavior of the RL agent may be triggered by poor definitions of the reward function for the given policy. Consider the limiting case where there are no trajectory errors, $e_i = 0$, and we do not include the count of crossings in the reward function, $w_{rev} = 0$. For a deterministic policy, the optimal expected return is null and the optimal policy is $\pi^* : ||\boldsymbol{\Delta V}|| = 0$. However, for an RL agent responding to an ϵ-greedy policy on an action space $||\boldsymbol{\Delta V}|| = [0,\, \Delta V_0]$, with schedule $\epsilon = \epsilon_0/k$ the expected return is

$$\mathbb{E}_{\pi^*}\{G_0 \,|\, \epsilon = \epsilon_0/k,\, ,\, w_{rev} = 0,\, e_i = 0\} = -\Delta V_0 \sum_{k}^{T} \frac{\epsilon_0}{k} \tag{19}$$

Recalling the integral test property for harmonic series, the right-hand side of Eq. (19) is bounded by

$$-\Delta V_0 \sum_{k}^{T} \frac{\epsilon_0}{k} < -\Delta V_0 \epsilon_0 \ln(T+1) \tag{20}$$

It follows from Eq. (20) that a necessary condition for the ϵ-greedy policy to converge to the optimal deterministic policy $\pi^* : ||\boldsymbol{\Delta V}|| = 0$ is

$$\epsilon_0 \ln(T+1) < 1 \tag{21}$$

The condition in Eq. (21) ensures that, under this example assumptions, the expected cost (i.e., a negative return) of possible random actions for T steps is less than the cost of a single action in a direction that terminates the episode after the first step.

Action Selection

The RL agent actions are orbit correction maneuvers, that are rendered by the velocity variation vector $\boldsymbol{\Delta V}_i$. The orbit correction maneuvers space is a real, continuous coordinate space of three dimensions, $\mathbb{R}^3$. Similar to the state feature space, the action feature space for the RL agent is constructed by aggregation (i.e., approximating the action value with zero-order polynomials on equally-spaced intervals Δa). To design effective RL agents it may be important to understand how $\boldsymbol{\Delta V}$ maneuvers impact the trajectory error that is incorporated into the reward function. We can quite rapidly identify such relationship if we assume a linear propagation of the error vector, i.e., $\boldsymbol{\delta}_{i+1} = \Phi\boldsymbol{\delta}_i$, where Φ is the state transition matrix defined in Eq. (3). Two relevant conditions are discussed for the linear case: 1) actions that null one of the final error vector components; 2) actions that minimize the norm of the final error. Actions that null the j-th component, δ_j, of the error vector, must satisfy

$$\delta_{i+1} = \Phi_\delta\boldsymbol{\delta}_i + \Phi_{\delta,V}\boldsymbol{\Delta V}_i = 0 \tag{22}$$

where Φ_δ is the row of the STM corresponding to the final error component δ_{i+1}, and $\Phi_{\delta,V}$ is the sub-block of Φ_δ that describes final variations due to an initial velocity change. Actions that null the j-th error component, are, therefore, a linear function of the initial error $\boldsymbol{\delta}_i$. Equation (22) yields an underdetermined system of equations for a complete velocity variation vector $\boldsymbol{\Delta V}_i$. This ambiguity may be removed by solving for the minimum norm solution of Eq. (22), or by assuming that two components of the maneuver vector $\boldsymbol{\Delta V}_i$ are null. Actions that minimize the final error norm also minimize the square of the norm

$$\boldsymbol{\delta}_{i+1}^T\boldsymbol{\delta}_{i+1} = \boldsymbol{\delta}_i^T\Phi^T\Phi\boldsymbol{\delta}_i + \boldsymbol{\Delta V}_i^T\Phi_V^T\Phi_V\boldsymbol{\Delta V}_i + 2\boldsymbol{\delta}_i^T\Phi^T\Phi_V\boldsymbol{\Delta V}_i \tag{23}$$

where Φ_V denotes the STM sub-block corresponding to initial velocity variations, and must satisfy

$$\nabla(\boldsymbol{\delta}_{i+1}^T\boldsymbol{\delta}_{i+1}) = 2\Phi_{\boldsymbol{V}}^T\Phi_{\boldsymbol{V}}\boldsymbol{\Delta V}_i + 2\Phi^T\Phi_{\boldsymbol{V}}\boldsymbol{\delta}_i = \mathbf{0} \tag{24}$$

Equation (23) may also be useful to estimate the range of maneuver size, $I_{\Delta V}$, that yields a contraction of the error norm

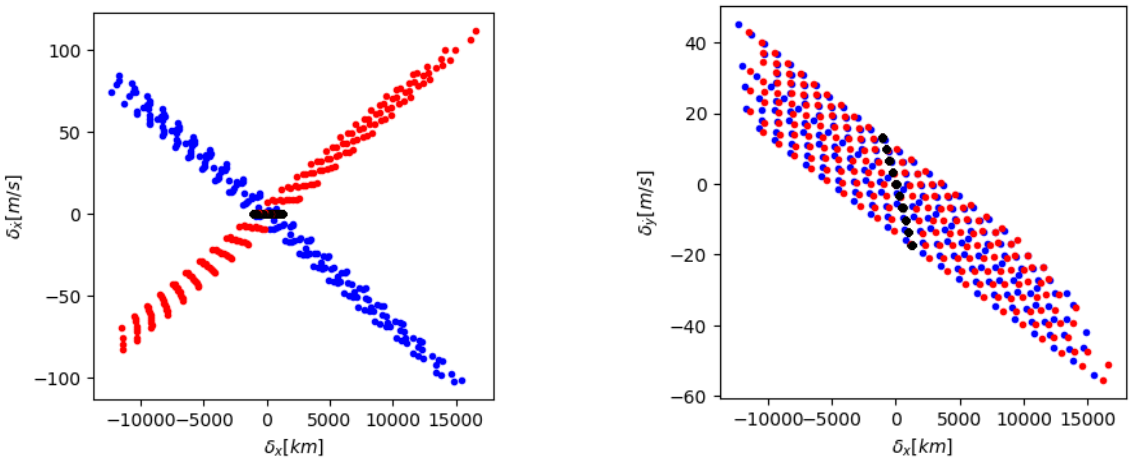
$$I_{\Delta V} : \left\{ \boldsymbol{\Delta V}_i^* \quad | \quad \boldsymbol{\delta}_{i+1}^T\boldsymbol{\delta}_{i+1} = \boldsymbol{\delta}_i^T\Phi^T\Phi\boldsymbol{\delta}_i + (\boldsymbol{\Delta V}_i^*)^T\Phi_{\boldsymbol{V}}^T\Phi_{\boldsymbol{V}}\boldsymbol{\Delta V}_i^* + 2\boldsymbol{\delta}_i^T\Phi^T\Phi_{\boldsymbol{V}}\boldsymbol{\Delta V}_i^* < \boldsymbol{\delta}_i^T\boldsymbol{\delta}_i \right\} \tag{25}$$

The range defined by Eq. (25) may be employed to estimate the size of the action discretization interval, Δa, or to determine the probability of ϵ-greedy actions to discover a better policy.

As for the construction of state features, the manifold topology near the reference orbit may guide the selection of RL agent action features and reduce the dimension of the action feature space. For example, consider a L_1 Lyapunov orbit with period $P = 15.48394714$ days in the Earth-Moon system ($\mu = 0.012151$). The manifold topology in vicinity of that reference orbit is displayed in Figure 3. The horizontal axis denotes the displacement along the x-axis relative to the reference orbit. The vertical axis denotes velocity component variations relative to the reference orbit. Black markers indicate sample adjacent Lyapunov orbits. Red markers describe sample points on nonlinear unstable manifolds that are associated with the selected range of Lyapunov orbits. Blue markers describe sample points on nonlinear stable manifolds that are associated with the selected range of Lyapunov orbits. From a visual comparison of Figure 3a and Figure 3b, it is apparent that changing velocity along the $\delta_{\dot{x}}$ direction allows a better control of the state location relative to the orbit manifolds than variations along the $\delta_{\dot{y}}$ direction. Therefore, the component of the maneuver vector $\boldsymbol{\Delta V}$ in the $\hat{x}$ direction may be sufficient to maintain the desired Lyapunov motion. On different families of reference orbit, similar considerations may offering a reduction of the action space dimension.

CASE STUDY RESULTS

The application of RL strategies to the stationkeeping of USP orbits is explored through a case study. L1 Lyapunov orbits are representative family of planar USP orbit behavior. For this case study we select a planar L1 Lyapunov orbit within the Earth-Moon system ($\mu = 0.012151$) with period $P = 13.08280257$ days. The analysis techniques applied to the selected L1 Lyapunov orbit may be replicated on any other USP orbit.



(a) $\delta_{\dot{x}}$ projection

(b) $\delta_{\dot{y}}$ projection

Figure 3: Representative manifold structure of adjacent orbits.

Insight from this study may facilitate subsequent investigations of different reference orbits or more complex applications.

Different property combinations, including distribution of initial conditions, state features, type of transitions, and reward definition lead to the creation of several RL agent configurations. Options for RL properties that are investigated in this work are listed in Table 1. This case study focuses on a foundational stationkeeping problem for USP orbits: given an initial perturbed state at the intersection with the x-axis, target a correction of the trajectory error at the successive crossing. Accordingly, RL agent state transitions propagate the orbit state from one crossing to the next, and simulation episodes are terminated after one, single step. The reward function is only determined by the trajectory error, e_i (and $w_e = 1$), with no contribution from the maneuver size ($w_{\Delta V} = 0$) or number of steps ($w_i = 0$). The parameter N_0 that delays the contraction of the epsilon value is empirically set to 100 for all simulations. Manifold geometry considerations as in Figure 3 are employed reduce the dimensionality of the action feature space, and only velocity variations in the x-axis direction are allowed. To also reduce the dimensionality of the state feature space, it is arbitrarily decided not to apply any initial trajectory error to the x component of the spacecraft orbit state, which is always nominal. All possible RL agent configurations are reduced to a representative subset which is listed in Table 2. Note that the RL agent label is formed by combining RL agent property abbreviations in the following order: type of initial conditions, type of state aggregation, type of transitions, and type of rewards. Performance of the RL agents listed in Table 2 are discussed in the following sections.

Convergence

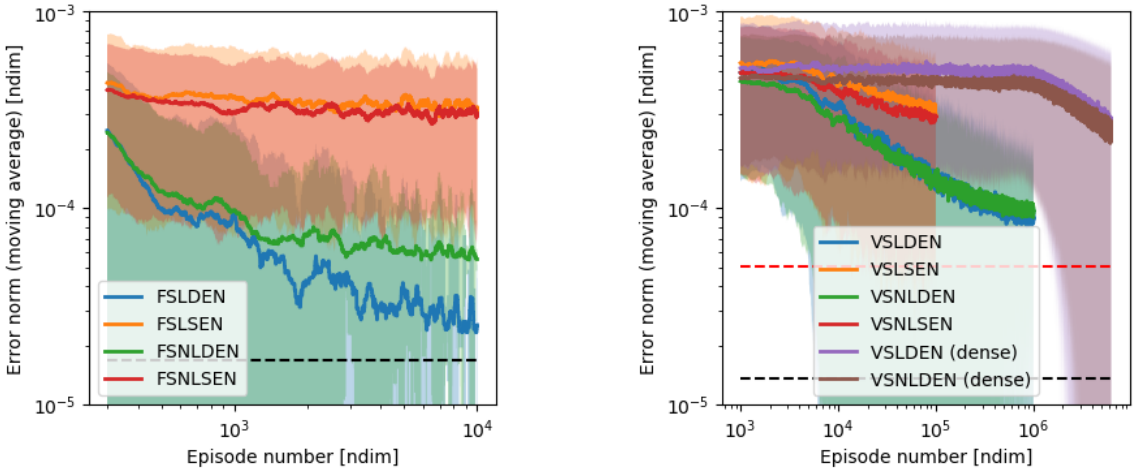
Training RL agents requires extensive computational time and resources. Figure 4 visualizes the convergence history for representative RL agents. Convergence is captured by the trajectory error norm (as defined in Eq. (16)). Results in Figure 4a are presented using a moving average with a window of 300 samples; in Figure 4b the moving window size is 500 samples. The black dashed line denotes the expected minimum error predicted for linear dynamics, the red dashed line the one predicted for fully nonlinear dynamics. Shaded areas represent the moving standard deviation over the sample window for each RL agent; While unambiguous convergence is not reached by either of the selected agents, convergence trends toward a minimum trajectory error value appear around 10^4 training episodes for fixed initial conditions, and $10^5 - 10^6$ for variable initial conditions. For comparison, when a dynamical model is provided, shooting algorithms may determine individual stationkeeping maneuvers in 4-100 Newton-Raphson iterations (assuming the Jacobian of the equations of motion is available), and a sequential quadratic programming optimization algorithm may

Table 1: RL agent properties.

Property	**Type (Abbreviation)**	**Description**
Initial conditions	fixed (F)	initial state is given and fixed in all components
	variable (V)	initial state components are drawn from independent normal distributions ($\sigma = 1$ km for y coordinate, $\sigma = 1$ cm/s for $\dot{x}$, and $\dot{y}$ coordinates)
State features	state aggregation (S)	state features created using Eq. (13) with $\delta_{\max} = -\delta_{\min} = 3$ km for position variables, $\delta_{\max} = -\delta_{\min} = 3$ cm/s for velocity variable
	manifold aggregation (M)	state features created using Eq. (14) with $\delta_{\max} = -\delta_{\min} = 3$ km
Number of state aggregation intervals	default	$n_s = 6$
Transitions	linear (L)	state propagation using state transition matrix ϕ
	nonlinear (NL)	state propagation by solving fully nonlinear Eq. (1)
	deterministic (D)	spacecraft state is fully known by agent
	stochastic (S)	spacecraft state is known by the agent with uncertainty $\sigma = 1$ km on the y coordinate and $\sigma = 1$ cm/s on the $\dot{x}$, and $\dot{y}$ coordinates
Rewards	error only:	$w_{\Delta V} = w_{rev} = 0$
	error norm (EN)	$e_i = \|\|\boldsymbol{\delta}_i\|\|$
	perpendicular crossing (XD)	$e_i = \|\delta_{\dot{x},i}\|$
	manifold (M)	$e_i = \|\|M^p \boldsymbol{\delta}_i\|\|$ with $p = 2$
Actions	ΔV_x	Maneuver in x-axis direction only, $\boldsymbol{\Delta V} = [\Delta V_x,\, 0,\, 0]^T$
Number of action aggregation intervals	default	$n_a = 50$

Table 2: RL agent configurations.

RL Agent	**Initial Conditions**	**State Aggregation**	**Actions**	**Transitions**	**Rewards**
FSLDEN	fixed (F)	state (S)	ΔV_x	L/D	error norm (EN)
FSNLDEN	fixed (F)	state (S)	ΔV_x	NL/D	error norm (EN)
FSLSEN	fixed (F)	state (S)	ΔV_x	L/S	error norm (EN)
FSNLSEN	fixed (F)	state (S)	ΔV_x	NL/S	error norm (EN)
VSLDEN	variable (V)	state (S)	ΔV_x	L/D	error norm (EN)
VSNLDEN	variable (V)	state (S)	ΔV_x	NL/D	error norm (EN)
VSLSEN	variable (V)	state (S)	ΔV_x	L/S	error norm (EN)
VSNLSEN	variable (V)	state (S)	ΔV_x	NL/S	error norm (EN)
VSLDEN (dense)	variable (V)	state (S) ($n_s = 50$)	ΔV_x ($n_a = 50$)	L/D	error norm (EN)
VSNLDEN (dense)	variable (V)	state (S) ($n_s = 50$)	ΔV_x ($n_a = 50$)	NL/D	error norm (EN)
VSLDXD	variable (V)	state (S)	ΔV_x	L/D	perpendicular crossing (XD)
VSLDM	variable (V)	state (S)	ΔV_x	L/D	manifold (M)
VMLDEN	variable (V)	manifold (M)	ΔV_x ($n_a = 6$)	L/D	error norm (EN)
VMLDEN (dense)	variable (V)	manifold (M) ($n_s = 50$)	ΔV_x ($n_a = 50$)	L/D	error norm (EN)
VMLDEN (denser)	variable (V)	manifold (M) ($n_s = 400$)	ΔV_x ($n_a = 400$)	L/D	error norm (EN)
VMLDM (dense)	variable (V)	manifold (M) ($n_s = 50$)	ΔV_x ($n_a = 50$)	L/D	manifold (M)



(a) Fixed initial conditions.

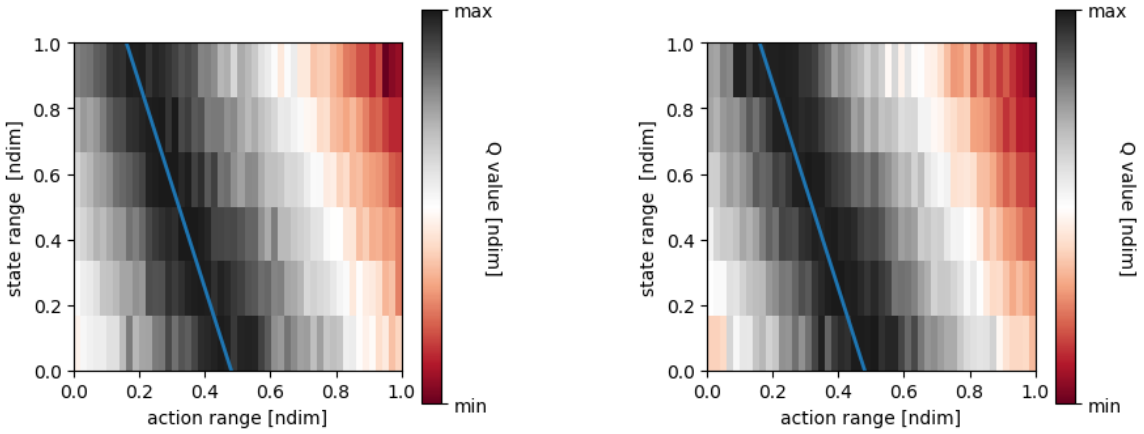
(b) Variable initial conditions.

Figure 4: RL agent convergence.

be able to identify a solution within 100-300 function evaluations (assuming no information on the Jacobian). We also note that in scenarios with dense state aggregation, such as for RL agent VSLDEN (dense) and VSNLDEN (dense), no improvement of the trajectory error occurs until a very large number of training episodes, e.g., 10^6. For a Monte Carlo control algorithm with state aggregation such behavior is expected. Ideally, the RL agent would need to visit each state-action pair at least once for a meaningful policy to start to emerge. Therefore, the number of episodes required before an actual policy improvement occurs is directly related to the dimension of the state-action space in a fundamental manner.

State-Action Value Function

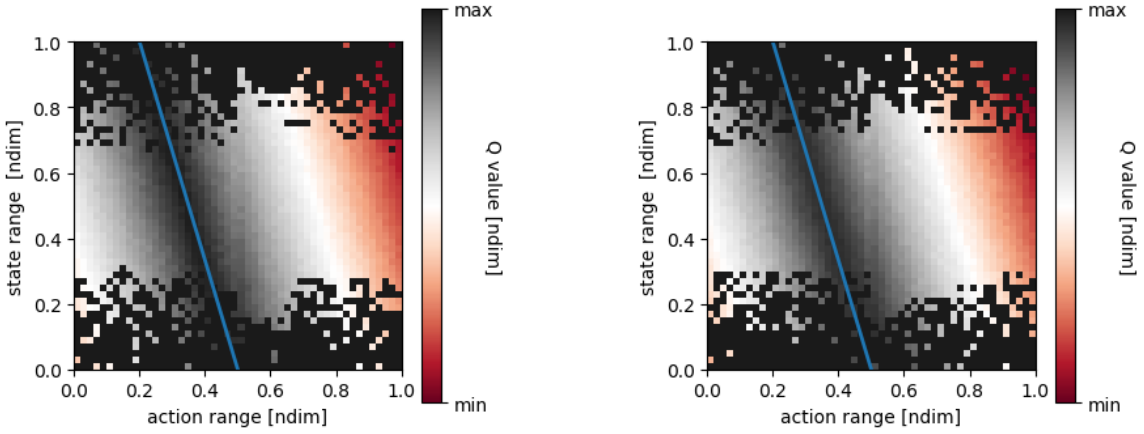
For the current RL agent configurations, orbit correction maneuvers are selected to maximize the state-action value function at the current agent state. To validate the results of the learning process we compare the topology of the resulting state-action value function with the location of local maxima that are predicted by linear theory. Figure 5 portrays such comparison for RL agent VSLDEN. The comparison is created by taking a section of the state-action value function at the half range for the state features δ_y and $\delta_{\dot{x}}$. Then, Figure 5 displays the variation of state-action value function over variations of the state features $\delta_{\dot{y}}$ and action feature ΔV_x. For given δ_y, $\delta_{\dot{x}}$, and $\delta_{\dot{y}}$ values, linear optimal maneuvers ΔV_x, may be predicted using Eq. (24). The location of linear optimal actions is marked by a blue line that overlaps the state-action value function surface in Figure 5. For linear transitions (see Figure 5a), the predicted location of linear optimal actions precisely matches the local maxima of the state-action value function, and confirms the potential ability of RL agents to learn optimal stationkeeping policies. As expected, the local maxima of the value function are offset relative to the location of linear optimal actions, when transitions between successive states are nonlinear. However, such offest is minimal for the selected RL agent configuration and reference orbit (see Figure 5b). As state aggregation becomes denser, the representation of the state-action value function may improve in certain regions of the state-action space. One challenge in increasing the number of intervals for state and action feature aggregation is that certain state-action pairs may remain unexplored during the learning process. In Figure 5 and 6, black pixels represent state-action pairs that are not encountered during training. The control policy remains undetermined at locations of the state-action space that are not visited by the RL agent. Alternative representations of the state space, including tiling,[17] may offer a better coverage of the control policy over the solution space.



(a) Linear transitions.

(b) Nonlinear transitions.

Figure 5: Section of state-action value function.



(a) Linear transitions.

(b) Nonlinear transitions.

Figure 6: Section of the state-action value function for denser state aggregation.

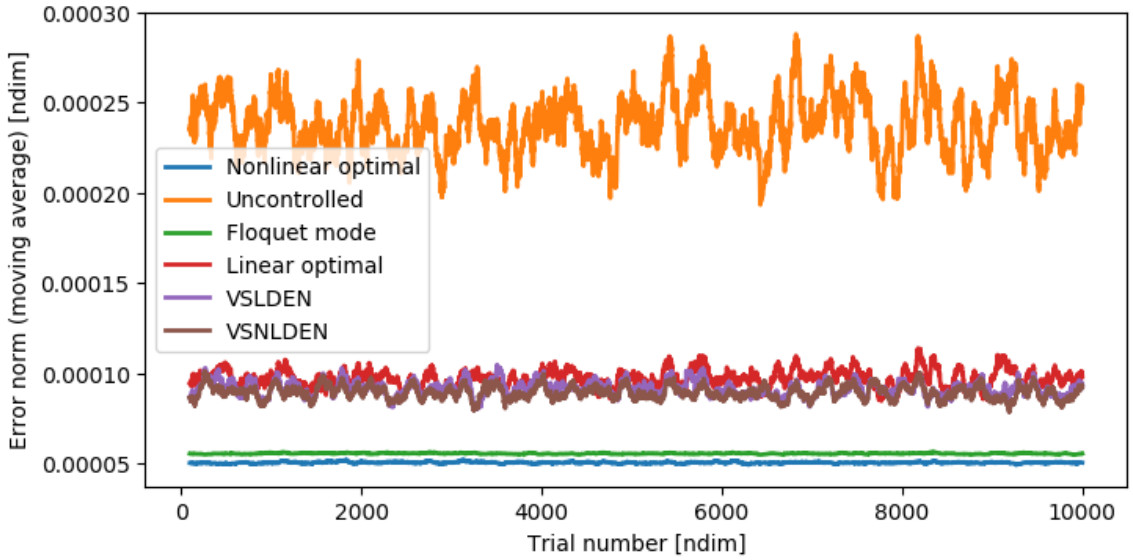


Figure 7: Error norm at half revolution for representative RL agents and existing stationkeeping strategies.

Trajectory Error

After training, RL agent performance are evaluated using a Monte Carlo approach. At each Monte Carlo simulation, randomly assign the agent initial state from a distribution identical to that employed during training. Then, compute and apply an orbit correction maneuver. Finally propagate the trajectory to next crossing of the x-axis and terminate the simulation. For this analysis, we assume RL agents to have fully accurate knowledge of their initial state, so that transitions are precisely deterministic. Simultaneous to velocity adjustments computed by a trained RL agent, alternative orbit correction maneuvers are determined via Floquet mode and optimal control for comparison. Optimal control is implemented in two forms: assuming linear transitions and solving Eq. (24), or assuming nonlinear transitions and computing the optimal solution using sequential quadratic programming. The objective for optimal control, in both forms, is the minimization of the error norm as defined in Eq. (16). The performance of each stationkeeping algorithm is first evaluated in terms of the final trajectory error norm. The trajectory error norm for representative RL agents and reference stationkeeping strategies is displayed over the sample of Monte Carlo trials in Figure 7. Each curve in Figure 7 denotes the moving average with window of 100 samples. The following is observed from Figure 7:

- Application of the selected RL agents results into a reduction of the trajectory error relative to the uncontrolled solution.
- Trajectory error reduction achieved through the selected RL agents is comparable to that obtained via linear optimal control.
- The selected RL agent that is trained assuming linear transitions performs similarly to the selected RL agent that is trained on a nonlinear dynamical model.
- Floquet mode control offers a performance in terms of trajectory error that is closest to the optimal solution among the stationkeeping strategies tested.

We also consider the trajectory error norm assuming that orbit dynamics are propagated from the terminal conditions for an additional orbit revolution. That corresponds to a trajectory error at one and half revolution and provides an additional metrics to evaluate stationkeeping performance. This metrics may enable to capture long-term effects of correction maneuvers on the orbit. Trajectory error norm at one and half revolutions over different Monte Carlo trials is reported in Figure 8, using a moving average with window of 100

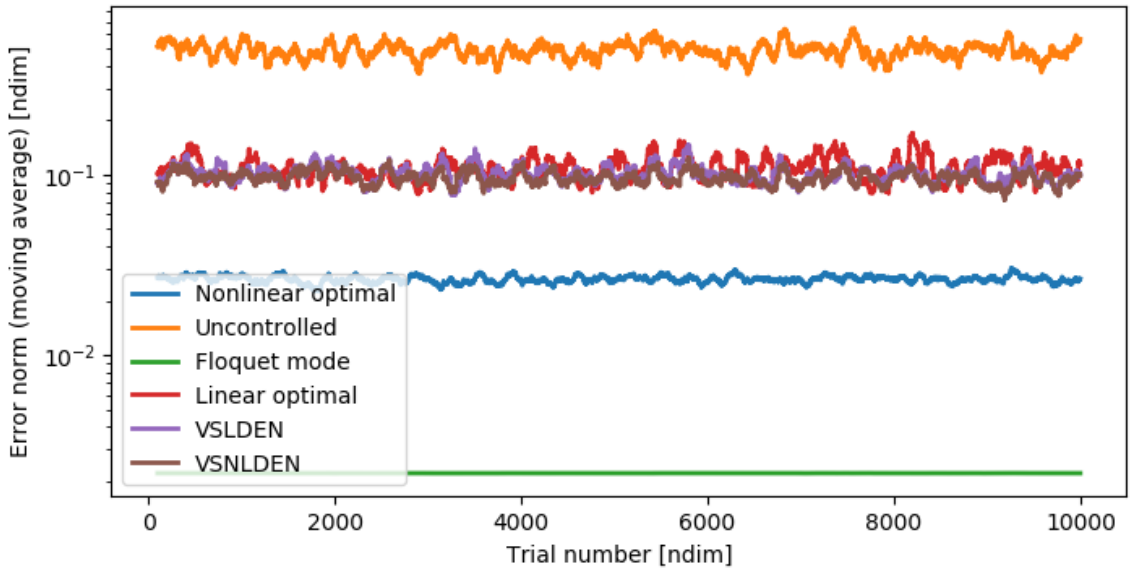


Figure 8: Error norm at one and half revolution for representative RL agents and existing stationkeeping strategies.

samples. It is apparent from Figure 8 that correction maneuvers that are precisely optimal on the short-period (i.e., half revolution) do not yield the best performance on a longer orbit propagation. On the long-period, Floquet mode control offers the smallest trajectory error norm among the strategies tested. That should not come as a surprise, as Floquet mode control aims to correct the asymptotic behavior of perturbed trajectories using orbit manifold information. Performance on the long-period in terms of trajectory error norm for the selected RL agents remains comparable to that for linear optimal correction maneuvers.

Comparison of Performance

This section presents the results of Monte Carlo simulations for all representative RL agents that are listed in Table 2. RL agent performance are evaluated based on three metrics: nominal trajectory error norm, maneuver size, and trajectory error norm at one and half orbit revolution. Metrics statistics for each agent are displayed through Figure 9 to 11. Black marks represent the average of the selected metrics over the Monte Carlo sample, black bars denote the corresponding standard deviation, and green bars indicate the corresponding min-max range. Referring to trajectory error norm statistics in Figure 9:

- All selected RL agents reduce the average trajectory error relative to uncontrolled motion.
- On average, nonlinear optimal control and Floquet mode control guarantee a smaller trajectory error than the selected RL agents.
- RL agents have larger variance of performance, and occasionally perform better than Floquet mode control and equally to nonlinear optimal control.
- Average performance for RL agents VSLDEN, VSNLDEN, VSLDXD, and VSLDM are comparable to the performance of a linear optimal controller.
- Increasing the number of state aggregation intervals for RL agents VSLDEN (dense) and VSNLDEN (dense) worsen the average performance. That may be a consequence of insufficient exploration of the state-action space during training.

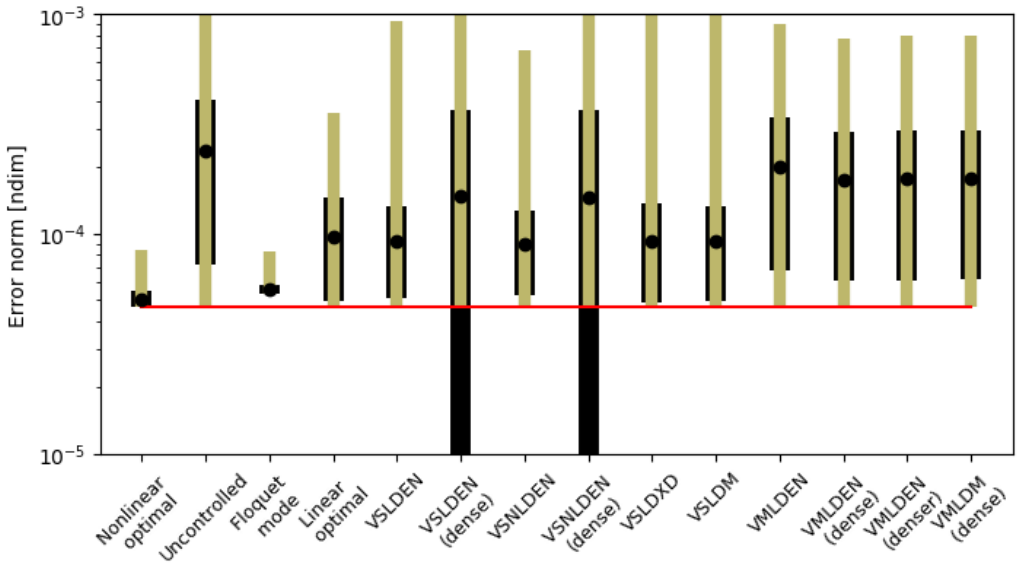


Figure 9: Statistics for error norm at half revolution for representative RL agents and existing stationkeeping strategies.

- Targeting perpendicular crossing (i.e., RL agent VSLDXD) or the p-orbit downstream trajectory error yields performance similar to agents that target error norm, but larger variance.
- Using manifold state aggregation reduces the dimensionality of the RL agent state feature space and allows for more aggregation intervals, but does not seem to improve the trajectory error for the selected configurations.

Maneuver size, $||\boldsymbol{\Delta V}||$, statistics are portrayed in Figure 10. According to Figure 10, maneuver size average and standard deviation for the selected RL agents and traditional stationkeeping algorithms are comparable. Min-max range is the largest on optimal control and Floquet mode control and the smallest on RL agents that adopt manifold state feature aggregation. Statistics for trajectory error norm at one and half orbit revolution are portrayed in Figure 10. It is apparent from this figure that Floquet mode control guarantees a significantly lower trajectory error on longer-term propagation than the other algorithms we tested. Recall that, for this analysis, orbit dynamics are precisely deterministic. In conclusion, this case study for deterministic orbit propagation shows that RL agents applied to stationkeeping of multi-body orbits display promising performance. However, improvement of RL agent configuration and training approach is warranted to achieve trajectory correction capabilities at the level of nonlinear optimal control and Floquet mode control.

Effect of Stochastic Transitions

The inclusion of stochastic effects into orbit propagation may hinder the efficacy of orbit maintenance algorithms and degrade their performance relative to deterministic dynamics. Such effects are magnified by the high sensitivity to small variations in the initial conditions that characterizes multi-body orbit dynamics. Under the assumption of deterministic dynamics, injecting dynamical model information into orbit maintenance algorithms (e.g., Floquet mode control utilizes known orbit manifolds and optimal control employees a known dynamical model) proves to be highly effective in reducing the trajectory error. Different behavior is observed when state transitions are stochastic. Stochastic transitions are modelled by incorporating a random displacement in spacecraft position and velocity after the application of a correction maneuver. Position errors are described by a normal distribution with standard deviation $\sigma = 1$ km; velocity errors are described by a normal distribution with standard deviation $\sigma = 1$ cm/s; Trajectory error norm at half revolution for

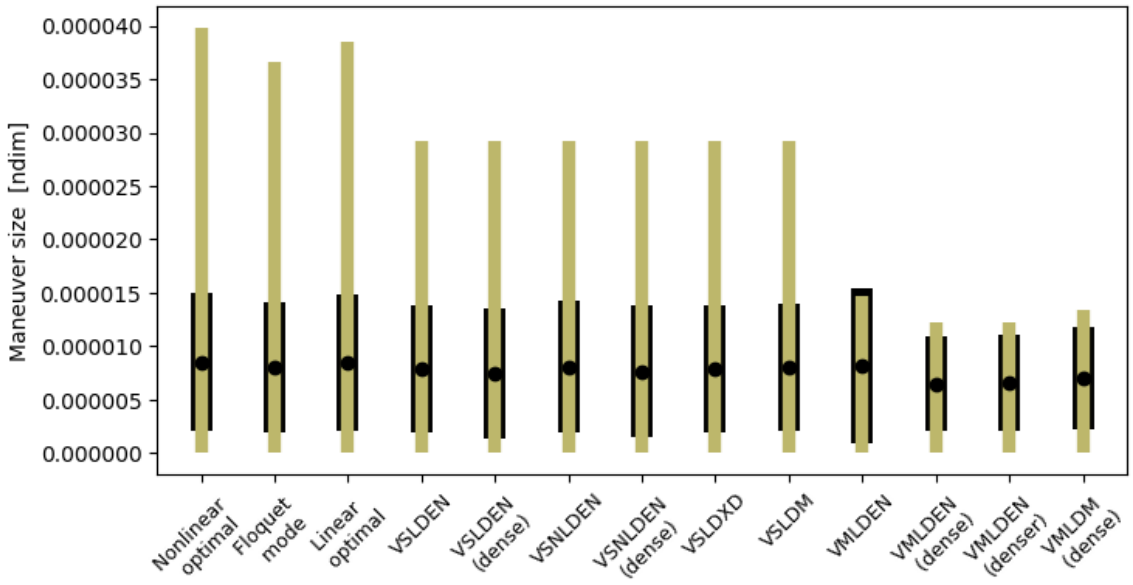


Figure 10: ΔV cost for representative RL agents and existing stationkeeping strategies

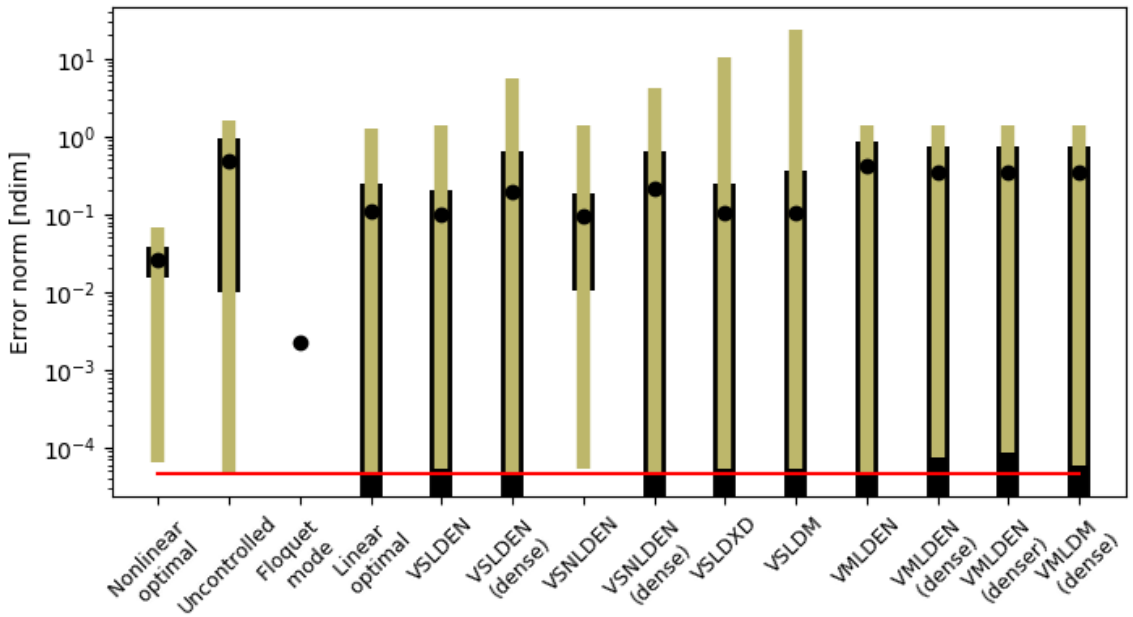


Figure 11: Statistics error norm at one and half revolution for representative RL agents and existing station-keeping strategies.

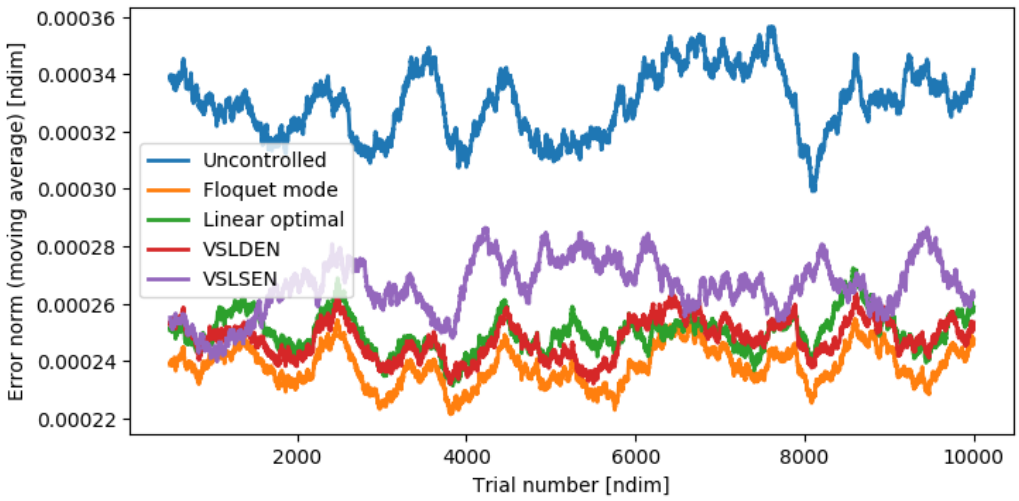


Figure 12: Error norm at half revolution for stochastic transitions.

a sample of Monte Carlo simulations and representative stationkeeping algorithms is plotted in Figure 12. Trajectory error norm is displayed as moving average on a window of 500 samples. Correction maneuvers are still generally beneficial within stochastic dynamics, as they significantly reduce the trajectory error relative to uncontrolled spacecraft motion. However, most of the advantages of incorporating dynamical model information into the orbit maintenance algorithm appear lost. In fact, trajectory errors for the selected RL agents and Floquet mode control are now comparable in Figure 12 (stochastic dynamics) as oppose to Figure 7 (deterministic dynamics). RL agents may be trained to identify an optimal policy under the assumption of stochastic transitions. That offers, in principle, an advantage over traditional stationkeeping approaches. Figure 12 includes the results for an RL agent, i.e. RL agent VSLSEN, that is trained over stochastic transitions. However, the trajectory error achieved by RL agent VSLSEN is no smaller than that achieved by RL agent VSLDEN, which is trained over deterministic transitions. A different agent configuration or a longer learning process may be warranted for RL agent VSLSEN to improve its performance. Within multi-body dynamics, stochastic behavior, such as that stemming from orbit determination errors, may equalize trajectory error performance for the selected stationkeeping algorithms. With further research, RL algorithms may offer an opportunity to improve orbit maintenance capabilities within stochastic regimes of motion.

Final Remarks

In this paper we explore the application of reinforcement learning to stationkeeping of unstable symmetric periodic orbits, a class of multi-body trajectories. Inspired by recent development in robotics, reinforcement learning may be applied to devise spacecraft maneuver planning algorithms that, after training on ground within a simulation environment, retain learning and adaptation capabilities in flight for autonomous path-planning. In this work we discuss a tabular, Monte Carlo implementation of reinforcement learning for orbit maintenance and explore its performance on a fundamental stationkeeping problem, one that may serve as a benchmark for performance. We empirically studied the effects of different training conditions on convergence and trajectory error suppression. We show results for reinforcement agent training conducted on linear/nonlinear orbit dynamics, fixed/variable initial conditions, deterministic/stochastic transitions, and different form of state aggregation. Observations collected in this study may inform the formulation of future reinforcement learning algorithms for orbit maintenance within gravitational multi-body regimes.

REFERENCES

[1] K. Laurini, B. Hufenbach, N. Satoh, J. Hill, and A. Ouellet, "The global exploration roadmap and expanding human/robotic exploration mission collaboration opportunities," 2015.

[2] J. P. Gardner, J. C. Mather, M. Clampin, R. Doyon, M. A. Greenhouse, H. B. Hammel, J. B. Hutchings, *et al.*, "The James Webb Space Telescope," *Space Science Reviews*, Vol. 123, No. 4, 2006, pp. 485–606.

[3] R. Whitley and R. Martinez, "Options for Staging Orbits in Cislunar Space," Big Sky, Montana, IEEE, March 2016.

[4] D. C. Davis, S. M. Phillips, and B. P. McCarthy, "Trajectory Design for Saturnian Ocean Worlds Orbiters Using Multidimensional Poincaré Maps," *Acta Astronautica*, Vol. 143, No. (Feb.), 2018, pp. 16–28.

[5] D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Stationkeeping Analysis for Spacecraft in Lunar Near Rectilinear Halo Orbits," San Antonio, Texas, Aug. 2017.

[6] D. C. Folta, T. A. Pavlak, K. C. Howell, M. A. Woodard, and D. W. Woodfork, "Stationkeeping of Lissajous Trajectories in the Earth-Moon System with Applications to ARTEMIS," *AIAA/AAS Astrodynamics Specialist Conference*, Monterey, California, July 2010.

[7] National Research Council, *NASA space technology roadmaps and priorities: Restoring NASA's technological edge and paving the way for a new era in space*. Washington DC: National Academies Press, 2012.

[8] R. W. Farquhar, "The Utilization of Halo Orbits in Advanced Lunar Operations," NASA technical note NASA TN D-6365, 1971.

[9] E. D. Gustafson and D. J. Scheeres, "Optimal Timing of Control-Law Updates for Unstable Systems with Continuous Control," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 878–887.

[10] K. Tajdaran, "Incorporation of Mission Design Constraints in Floquet Mode and Hamiltonian Structure-Preserving Orbital Maintenance Control Strategies for Libration Point Orbits," Master's thesis, Purdue University, School of Aeronautics and Astronautics, 2015.

[11] K. C. Howell and T. M. Keeter, "Station-Keeping Strategies for Libration Point Orbits - Target Point and Floquet Mode Approaches," *Spaceflight mechanics*, Vol. 99, No. 2, 1995, pp. 1377–1396.

[12] C. Simó, G. Gómez, J. Llibre, R. Martinez, and J. Rodriguez, "On the Optimal Station Keeping Control of Halo Orbits," *Acta Astronautica*, Vol. 15, No. 6-7, 1987, pp. 391–397.

[13] G. Gómez, K. C. 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.

[14] K. Williams, R. Wilson, M. Lo, K. C. Howell, and B. Barden, "Genesis Halo Orbit Station Keeping Design," *International Symposium: Space Flight Dynamics*, Biarritz, France, June 2000.

[15] D. Rohrbaugh and C. Schiff, "Station-Keeping Approach for the Microwave Anisotropy Probe (MAP)," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, California, August 2002.

[16] D. W. Dunham and R. W. Farquhar, "Libration Point Missions, 1978–2002," *Libration point orbits and applications*, pp. 45–73, Singapore: World Scientific, 2003.

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

[18] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep Q-learning from Demonstrations," *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, February 2018.

[19] X. B. Peng, P. Abbeel, S. Levine, and M. v. d. Panne, "Deepmimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills," *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, 2018, pp. 143–161.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," *arXiv preprint arXiv:1509.02971*, 2015.

[21] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," *Advances in neural information processing systems*, Denver, Colorado, December 2000.

[22] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *International Conference on Machine Learning*, Beijing, China, June 2014.

[23] K. Howell and H. J. Pernicka, "Station-keeping Method for Libration Point Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 151–159.

[24] V. Szebehely, *Theory of orbits: the restricted problem of three bodies*. New Haven, CT: Academic Press, 1967.

[25] D. Grebow, "Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage," Master's thesis, Purdue University, School of Aeronautics and Astronautics, 2006.

[26] D. Guzzetti, N. Bosanac, A. Haapala, K. C. Howell, and D. C. Folta, "Rapid Trajectory Design in the Earth–Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Orbits," *Acta Astronautica*, Vol. 126, 2016, pp. 439–455.