Apprenticeship Learning for Maneuver Design in Multi-Body Systems

Ian Elliott*, Natasha Bosanac†, Nisar Ahmed‡ , and Jay McMahon†
University of Colorado Boulder, Boulder CO, 80309

Although orbital maneuver design currently relies heavily on human flight dynamicists, autonomous maneuver planning technologies may enable missions that require a rapid response and support resiliency in uncertain environments. However, it is often challenging to precisely translate the maneuver design process, balancing near- and long-term objectives with various constraints, into a single analytical expression. Techniques based on inverse reinforcement learning offer one approach to approximately uncover the objectives of a maneuver planner. In this preliminary analysis, apprenticeship learning via inverse reinforcement learning is used to recover the strategy of a straightforward controller in the design of station-keeping and rendezvous maneuvers for a spacecraft in a near rectilinear halo orbit in cislunar space.

I. Introduction

The current state-of-the-art in spacecraft maneuver planning relies heavily on human-in-the-loop design and verification of heavily constrained solutions. During station-keeping and proximity operations, orbital maneuvers are regularly designed to achieve a variety of near- and long-term objectives, while also satisfying path, thrust plume, collision and other constraints. The design of maneuvers and associated trajectories that satisfy these constraints is especially challenging in a multi-body gravitational environment, where complex and chaotic dynamical systems govern the motion of spacecraft. As a result, these maneuver design activities are typically performed by a team of flight dynamicists. Autonomous spacecraft operations, which reduce the human-in-the-loop dependencies, may be invaluable for designing both regular maneuvers to station-keep and unexpected maneuvers in response to hazards or changing goals. In fact, autonomy in these decision-making activities may enhance a variety of missions by reducing operational complexity (e.g., for the components of space-based infrastructure or large swarms) and enable missions that necessitate rapid responses or suffer from time delays (e.g., in-space assembly or distant operations). One approach to autonomously planning robust, efficient and safe maneuvers is to mimic a human flight dynamicist who is able to reason, learn, and adapt. However, during the trajectory design process, it is challenging for a flight dynamicist to precisely translate the goals and constraints considered during maneuver planning into a single, analytical expression. To address these challenges, inverse reinforcement learning (IRL) – also referred to as inverse optimal control [1, 2] – and apprenticeship learning are used in this paper to uncover an approximation of the objective of a maneuver planner.

Several IRL-based approaches exist and are differentiated by the information available to describe the expert policy. For instance, the IRL methods derived by Ng and Russell [3] and Abbell and Ng [4] assume the reward at each state to be a weighted linear combination of features derived from a set of preconstructed solutions. Implementations of IRL that assume the reward is a linear combination of features are typically solved using a linear programming (LP) or quadratic programming (QP) formulation [3, 4]. These IRL methods enable a maximum margin based optimization, whereby the reward uncovered from an expert policy is an improvement over the reward uncovered from all non-expert policies [1, 5, 6]. Maximum margin IRL has previously been used in astrodynamics problems to recover the behavior of space objects by Linares and Furfaro [7], specifically to detect maneuvers and estimate the associated Δv for space objects in a geostationary orbit. In addition, entropy-based IRL methods, that maximize the expected entropy given by a policy have also been presented and applied by Ziebart, Maas, Bagnell and Dey and Fu, Luo and Levine [8, 9]. This maximum entropy IRL formulation has been used by Doerr, Linares and Furfaro to estimate the behavior of space objects in low-Earth and geostationary orbits [10].

*Graduate Research Assistant, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, 429 UCB University of Colorado Boulder, Boulder CO, 80303

†Assistant Professor, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, 429 UCB University of Colorado Boulder, Boulder, CO 80303, AIAA Member.

‡Assistant Professor, Research & Engineering Center for Unmanned Vehicles, Smead Department of Aerospace Engineering Sciences, 429 UCB University of Colorado Boulder, Boulder, CO 80303, AIAA Member.

Apprenticeship learning is a related process to IRL: rather than directly recovering the reward function of an expert, apprenticeship learning recovers an estimated policy that produces trajectories closely resembling the trajectories produced by the expert policy [11]. In the apprenticeship learning process, an IRL algorithm is used as a critical step to iteratively update the estimated, or apprentice, policy until the value of the estimated policy converges to the value of the expert. Apprenticeship learning has been applied to learning the actions of an agent in a variety of scenarios and environments. One notable example is the recovery of driving styles from observing a human drive a car in a simulated environment; in such a scenario, no exact analytical and generalizable reward function may be defined to describe the intentions and decisions of the human [4]. Apprenticeship learning has also been applied to acrobatic helicopter maneuver design following expert helicopter pilot demonstrations of complex maneuvers [12, 13]. By examining an expert demonstration of a helicopter pilot, apprenticeship learning enables identification of an apprentice policy that successfully recreated a variety of helicopter maneuvers, mimicking a skilled human pilot.

This paper explores the use of apprenticeship learning to recover the objectives of a maneuver planner for a spacecraft in a chaotic multi-body gravitational environment using demonstrations from an expert policy; in this preliminary analysis, the expert policy is a straightforward continuous and unconstrained controller. In fact, expert trajectories are generated using a policy defined by a linear quadratic regulator (LQR) control model with a known cost function and associated gains; this controller is selected to support efficient trajectory generation and to enable straightforward verification of the results of the apprenticeship learning procedure. Then, the gains associated with the LQR controller are selected to design trajectories in two distinct maneuvering scenarios: rapid rendezvous and station-keeping. In addition, the uncontrolled dynamics of the spacecraft in the Earth-Moon system are modeled using the Circular Restricted Three-Body Problem (CR3BP), which sufficiently approximates the complex nature of the dynamical environment in cislunar space. In this dynamical model, a Near-Rectilinear Halo Orbit (NRHO) in the vicinity of L_2 in the Earth-Moon system is selected as the reference orbit; this particular orbit is of much interest in the space community as a potential location for the upcoming Lunar Orbital-Platform Gateway [14]. Then, the expert policy defined by the LQR controller is used to design a controlled trajectory for a spacecraft in the vicinity of an L_2 NRHO reference orbit. The gains are selected to recover trajectories in two distinct maneuver scenarios: station-keeping and rendezvous. In each scenario, trajectories generated using this expert policy are input to the apprenticeship learning algorithm to recover an apprentice policy that produces similar solutions to the expert's policy. One by-product of apprenticeship learning is an estimate of the reward or objective function maximized by the apprentice policy; while this reward function is typically not a close approximation of the objective of the expert, it may provide sufficient insight into the priorities of the maneuver planner. In this paper, the outputs of this apprenticeship learning procedure are analyzed and compared to the LQR controller via the trajectories generated by the apprentice policy as well as the estimated reward function. The results of this analysis indicate that, in the context of this spacecraft maneuver planning scenario, apprenticeship learning generates policies that resemble the LQR controller and recovers the priorities of the expert to a similar order of magnitude.

II. Background

A. Dynamics Model

The CR3BP offers a suitable approximation of the natural dynamics governing the motion of a spacecraft in the chaotic multi-body gravitational environment of the Earth-Moon system. The CR3BP approximates the path of the two primary bodies – the Earth, P_1, and the Moon, P_2 – via circular orbits about their mutual barycenter. A spacecraft, P_3, is an assumed massless particle and is only influenced by the point-mass gravity of the two primary bodies. An Earth-Moon rotating coordinate system is defined using a frame $(\hat{x}, \hat{y}, \hat{z})$ and is centered at the system barycenter: $\hat{x}$ is defined in the direction of P_2, such that the bodies P_1 and P_2 are fixed along the $\hat{x}$ axis; $\hat{z}$ is defined in the direction of the angular velocity of the system; and $\hat{y}$ completes the right-handed coordinate system. The rotating frame is illustrated in Fig. 1, along with the Earth, Moon, and spacecraft, not to scale. A nondimensionalization scheme is also employed. Length quantities are normalized using the Earth-Moon distance, such that the nondimensional distance between the Earth and Moon is unity. Quantities of time are nondimensionalized such that the period of the Earth and Moon about their barycenter is 2π. Finally, mass quantities are nondimensionalized using the total mass of the Earth-Moon system and a mass ratio, μ, is defined as:

$$\mu = \frac{m_2}{m_1 + m_2} \tag{1}$$

where m_1 is the mass of the Earth and m_2 is the mass of the Moon. The nondimensional state vector of a spacecraft in the rotating frame relative to the system barycenter is then formulated as $\vec{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$. In this analysis, the natural

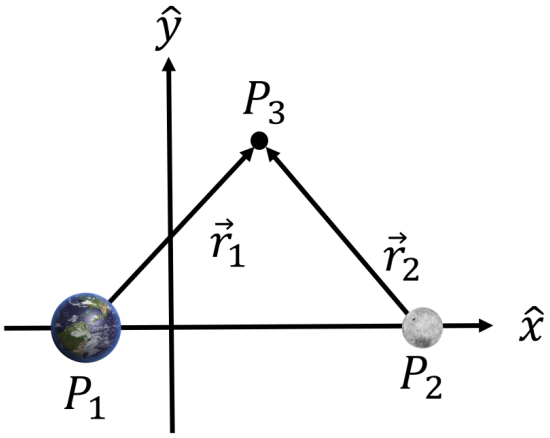


Fig. 1 Definition of the rotating frame in the Earth-Moon CR3BP.

motion approximated by the CR3BP is augmented by a control acceleration. This nondimensional acceleration vector is defined using the components u_x, u_y, and u_z, applied in the $\hat{x}$, $\hat{y}$, and $\hat{z}$ directions, respectively. Using these definitions, the controlled motion of the spacecraft is described by the following equations of motion:

$$\begin{aligned} \ddot{x} &= 2\dot{y} + x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} + u_x \\ \ddot{y} &= -2\dot{x} + y - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} + u_y \\ \ddot{z} &= -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} + u_z \end{aligned} \tag{2}$$

where $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ are the distances of P_3 from P_1 and P_2, respectively [15]. When $u_x = u_y = u_z = 0$, these equations of motion correspond to the CR3BP: the underlying dynamical structure admits a variety of fundamental solutions including equilibrium points and periodic orbits that may be approximately retained in higher fidelity models of the cislunar environment.

B. Reference Orbit

An NRHO is a periodic orbit that exists in the CR3BP along a subset of a halo orbit family associated with either L_1 or L_2. These orbits possess a low perilune and high apolune and are nearly polar. An Earth-Moon L_2 southern NRHO has been identified as a favorable option for near-future lunar operations due to favorable characteristics including neutral dynamical stability, availabilities of communications line-of-sights, and low Δv requirements [14, 16]. Specifically, an L_2 NRHO with a 9:2 lunar synodic resonance has been cited as a likely destination for the Lunar Orbital Platform-Gateway and serves as the reference orbit used in this analysis. This orbit, propagated using the dynamics of the CR3BP, is depicted in Fig. 2 in the Earth-Moon rotating frame with the L_2 equilibrium point plotted as a red diamond for scaling purposes. In this figure, the black arrow indicates the direction of motion along the NRHO. This NRHO is highly eccentric and highly inclined with respect to the Earth-Moon plane; the orbit also possesses a period of 6.062 days. Modeling the Moon as a sphere with a radius of 1738 km, the reference NRHO possesses a lunar periapsis altitude of approximately 294 km and an apoapsis altitude of approximately 66,390 km. Due to this large range in the distance to the Moon, the relative motion dynamics along the reference orbit vary significantly as the spacecraft travels along the NRHO.

C. Linear Quadratic Regulator Control

To define a metric for evaluating the results of the apprenticeship learning algorithm in the context of this preliminary analysis, an expert policy is created to maximize a known reward function. In this paper, LQR is used to design controlled trajectories around natural reference solution, defined as the Earth-Moon southern L_2 NRHO. LQR is a

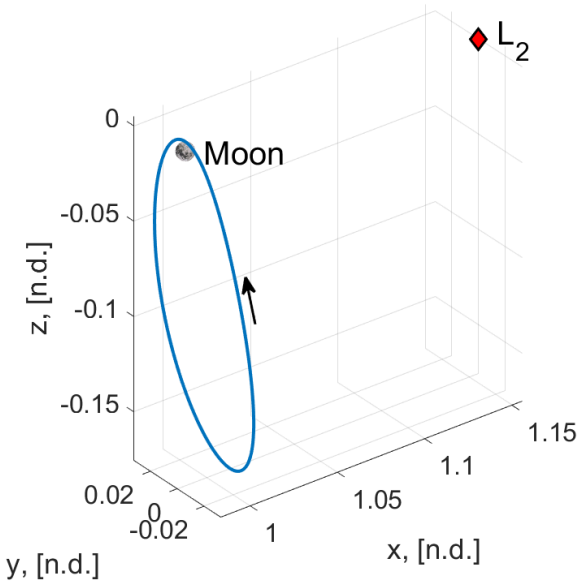


Fig. 2 Earth-Moon near rectilinear halo orbit associated with L_2 with a 9:2 lunar synodic resonant period in the rotating frame using nondimensional coordinates.

suitable candidate for constructing an expert policy as the LQR objective function that is minimized by the controller is straightforwardly translated into a reward function that is optimized by the policy. The goal of LQR is to minimize the following cost or objective function, evaluated over a finite time horizon from an initial time, t_0, to a final time, t_f:

$$J = \int_{t_0}^{t_f} \delta\vec{x}^T Q \delta\vec{x} + \delta\vec{u}^T R \delta\vec{u}\, dt \tag{3}$$

where Q and R are gain matrices associated with the state error, $\delta\vec{x}$, and control error, $\delta\vec{u}$, respectively. These gain matrices must be adjusted to achieve a desired behavior of the spacecraft trajectory relative to the reference. For an n-dimensional state vector and p-dimensional control input, the Q and R matrices possesses dimensions of $n \times n$ and $p \times p$ respectively; upon augmenting the CR3BP with a control acceleration, $n = 6$ and $p = 3$. Then, the relative state, $\delta\vec{x}$, is defined relative to a state, $\vec{x}_r$, along the reference orbit via an isochronous correspondence, i.e., $\delta\vec{x}(t) = \vec{x}(t) - \vec{x}_r(\mathrm{t})$. Since the reference orbit in this analysis is defined as a natural trajectory, the control usage error is equivalent to the control acceleration for the maneuvering spacecraft, i.e., $\delta\vec{u} = \vec{u}$. The optimal control acceleration, $\vec{u}$, that minimizes the LQR cost function is calculated using the following state feedback-control law:

$$\vec{u} = -K\delta\vec{x} \tag{4}$$

where the time-varying matrix K is equal to:

$$K = R^{-1}BP \tag{5}$$

and the matrix B is the following input matrix:

$$B = \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix} \tag{6}$$

where $0_{3\times3}$ is a 3×3 matrix of zeros and $I_{3\times3}$ is a 3×3 identity matrix. Then, the matrix P is the solution to the Riccati differential equation, written mathematically as:

$$\dot{P} = -A^T P - PA + PBR^{-1}B^T P - Q \tag{7}$$

with the boundary condition $P(t_f) = 0_{n\times n}$. In this expression, the matrix A is the time-varying state derivative matrix evaluated along the reference trajectory, $\vec{x}_r$. Following definition of the Q and R matrices, this LQR controller is used to efficiently generate trajectories that minimize the LQR cost function over a defined finite time horizon.

D. Overview of Inverse Reinforcement Learning

Inverse reinforcement learning is the process of recovering the reward function that is maximized by an expert given observations of the solutions generated under the expert policy. In the context of the spacecraft maneuver planning scenario analyzed in this paper, the behavior of the expert policy is observed through a set of trajectories generated using an LQR controller. Application of this approach to the spacecraft maneuver planning problem is depicted conceptually in Fig. 3. First, a set of maneuvers and associated trajectories is generated via an LQR controller for initial conditions that are perturbed from the reference NRHO. These trajectories are sampled in time and input to an IRL algorithm along with a model of the CR3BP and the LQR controller. The goal of IRL is to then approximate the reward function that is maximized by the policy used to generate the input solution demonstrations. In this process, the input data set and algorithm formulation significantly influence the obtained reward function.

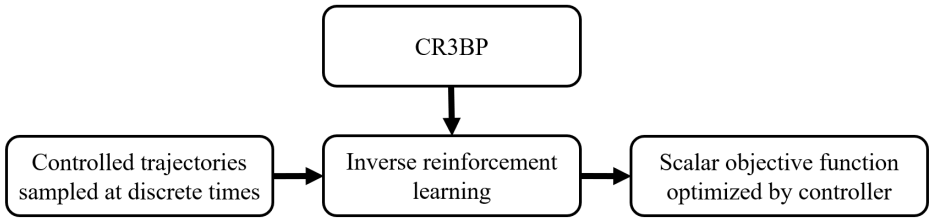


Fig. 3 Conceptual representation of the inverse reinforcement learning process used to recover the approximate reward from trajectories sampled from an expert policy.

The path followed by an agent in an environment that is described by a continuous state space and an associated set of actions is formulated as a continuous Markov decision process (MDP). This MDP is defined using the tuple $(S, A, P_{sa}, \gamma, D, R)$ where S is a set of continuous states, A is set of continuous actions, P_{sa} is the state transition probability of taking action a at state s, γ is the discount factor, defined as $\gamma \in (0, 1]$, D is the distribution of initial states, and R is the reward function. The reward function is designed to reward favorable actions and penalize unfavorable actions. A policy, π, is then defined as a set of mappings from the set of states S to the set of actions A. The expected value, $E[V^\pi]$, of a policy is expressed as:

$$E[V^\pi] = E\left[\sum_{i=0}^{\infty} \gamma^i R(\vec{x}_i, \vec{u}_i) \,\middle|\, \pi\right] \tag{8}$$

to reflect the expected sum of rewards over time along a trajectory [4]. Given trajectories generated by a policy and discretely sampled in time, a vector of features, $\vec{\phi}(\vec{x}, \vec{u})$, is defined using characteristics of the solutions and the objectives of the maneuver planner. The IRL algorithm implemented in this analysis assumes that the reward function $R(\vec{x}_i, \vec{u}_i)$ is a linear combination of the components of the feature vector evaluated at each time along a trajectory, scaled by a vector of weights, $\vec{w}$. Thus, the reward function at time t_i is expressed as $R(\vec{x}_i, \vec{u}_i) = \vec{w}^T \vec{\phi}(\vec{x}_i, \vec{u}_i)$. The feature expectations vector of a policy, $\vec{f}(\pi)$, is then defined as:

$$\vec{f}(\pi) = E\left[\sum_{i=o}^{\infty} \gamma^i \vec{\phi}(\vec{x}_i, \vec{u}_i) \,\middle|\, \pi\right] \tag{9}$$

Then, the expected value of the policy is related to the feature expectation vector via the following relationship:

$$E[V^\pi] = \vec{w}^T \vec{f}(\pi) \tag{10}$$

Using the feature expectations vector associated with an expert policy, labeled $\vec{f}_E$, the goal of IRL is to recover the weight vector, $\vec{w}_E$, associated with the reward function. However, a well-known challenge in IRL is the ambiguity associated with the recovered estimate of the reward function. There may exist several weight vectors that explain the expert's feature expectations vector; or, equivalently, the expert policy may maximize several reward functions [3].

One approach to approximating the expert's reward function is maximum margin IRL. This particular algorithm recovers a set of reward weights that correspond to the expert policy possessing a greater expected value than the expected value of other, non-expert policies [4]. This algorithm is formulated as an optimization problem by introducing

a scalar margin, β, between the expected value of the expert policy, and the expected value of a non-expert policy. Maximization of this scalar margin is written mathematically as:

$$\begin{aligned} &\text{Maximize } \beta \\ &\text{Such that } \vec{w}^T \vec{f}_E \geq \vec{w}^T \vec{f} + \beta \\ &\qquad |\vec{w}| \leq 1 \end{aligned} \tag{11}$$

where $\vec{f}$ is the feature expectations vector of the non-expert policy. The variables recovered via this optimization problem include both β and $\vec{w}$. In addition, the Euclidean norm of the weight vector is constrained to values less than or equal to unity to bound the search [4]. Since this optimization problem is composed of a scalar objective function, and both linear and quadratic constraints, it may be solved using a nonlinear optimization solver, e.g., an interior point algorithm. In this paper, MATLAB's *fmincon* algorithm is employed.

To assess the accuracy of the reward function recovered by the IRL algorithm, a straightforward transformation between the weight vector and LQR gains is constructed. First, to estimate the objective of the expert – in this case, an LQR controller – a reward function is constructed as a linear combination of nine features. Consider an LQR controller with a set of gain matrices, Q and R, that are diagonal. For this controller, the associated features are straightforwardly defined as the six state errors squared, and the three control inputs squared. Thus, the exact feature vector at each state is defined as:

$$\vec{\phi}(\delta\vec{x}, \vec{u}) = \begin{bmatrix} \delta x^2 & \delta y^2 & \delta z^2 & \delta \dot{x}^2 & \delta \dot{y}^2 & \delta \dot{z}^2 & u_x^2 & u_y^2 & u_z^2 \end{bmatrix}^T \tag{12}$$

Then, the exact reward function for an LQR controller is equivalent to:

$$R(\delta\vec{x}, \vec{u}) = \alpha \left[-\delta\vec{x}^T Q \, \delta\vec{x} - \vec{u}^T R \, \vec{u} \right] \tag{13}$$

where the scalar quantity α is employed to scale the Euclidean norm of the coefficients of the reward function to possess a magnitude that is less than or equal to unity [12]. Note that the reward function and cost function for an LQR controller differ in sign: the coefficients in the reward function are negative with a maximum scalar value of zero. Then, the components of the true weight vector $\vec{w}$ associated with this reward function are equal to the negative of each of the diagonal elements of the two gain matrices, scaled by α. Mathematically, this relationship is written as:

$$\vec{w} = -\frac{1}{\alpha} \begin{bmatrix} Q_x & Q_y & Q_z & Q_{\dot{x}} & Q_{\dot{y}} & Q_{\dot{z}} & R_{u_x} & R_{u_y} & R_{u_z} \end{bmatrix} \tag{14}$$

Conversely, to convert the estimated weights into an LQR controller model, the sign of the weights must first be flipped to be positive and multiplied by the scalar quantity α. The value for α is selected using the a priori known minimum or maximum value of the elements of the gain matrices.

E. Apprenticeship Learning

Apprenticeship learning is used to recover a policy that produces trajectories with similar characteristics to those generated by an expert policy [4]. Given a feature expectations vector evaluated using the expert policy, an apprenticeship learning algorithm estimates the weight vector associated with the reward function; the approach leveraged in this analysis incorporates maximum margin IRL, as summarized in Eq. 11. With the estimated weights, an estimated policy is generated using a given dynamics and controller model. Then, an estimated feature expectations vector is evaluated using a set of trajectories generated with the estimated policy and compared to the feature expectations vector associated with the expert policy. This process is repeated until the feature expectations vector derived from the estimated policy, or "apprentice" policy, closely matches the feature expectations vector of the expert policy. While this procedure is not guaranteed to recover the exact weights used to define the expert's reward function, the policy generated through apprenticeship learning resembles the expert's policy if the algorithm converges to a solution based on a specified set of termination criteria [4]. These steps of the apprenticeship learning process are summarized as follows:

1) Using trajectories generated via the expert policy, evaluate the expert feature expectations vector, $\vec{f}_E$. Define an initial guess for an apprentice policy and evaluate the corresponding feature expectations vector, $\vec{f}_0$.
2) Apply maximum margin inverse reinforcement learning as outlined in Eq. 11 to estimate the weight vector, $\vec{w}_i$, of the estimated reward function at iteration i for the specified feature basis functions.
3) Check if the margin, β, possesses a value below a cutoff tolerance, ϵ.

4) Using the estimated weights at iteration i to define the estimated reward function, generate the current apprentice policy, π_i.
5) Generate trajectories using π_i and sample at discrete times. Using these state sequences, calculate the associated features expectations vector, f_i.
6) Repeat steps 2-5 until either: convergence to a solution, as indicated by the value of β falling below a defined a cutoff tolerance; failure of the algorithm to converge, as defined by reaching a maximum number of iterations.

This general procedure is depicted conceptually in Fig. 4. During apprenticeship learning, it is necessary to compute the feature expectations vector along trajectories generated using both the expert and estimated policies. For both policies, the same process is used to generate and sample trajectories prior to computing the feature expectations vectors. First, initial state errors are randomly drawn from the same distribution relative to a state defined along the reference trajectory. The trajectory associated with each initial condition are generated using the specified dynamics and control model for the same duration and the trajectories are sampled at evenly-spaced time steps. Finally, the features expectations vector is calculated from the generated trajectories using Eq. 9. The number of generated trajectories, number of sampled states per trajectory, and discount factor are all tunable parameters for the computation of the feature expectations – and, therefore, influence the results of the apprenticeship learning algorithm. For all cases in this preliminary analysis where the dynamics and control models are deterministic, a discount factor of $\gamma = 1$ is selected. Then, the number of generated trajectories and the number of sampled states along each trajectory are each selected based on insight from previous studies of apprenticeship learning in other disciplines and a parameter search process conducted using the scenario of interest [4].

III. Applying Apprenticeship Learning to Controlled Spacecraft Trajectory Design

To demonstrate the capability for apprenticeship learning to estimate the reward function describing the LQR controller implemented in the CR3BP, three test cases are explored. In this particular application, the dynamics and control models are set equal to the CR3BP and the LQR controller, respectively. Then, generating a new policy from an estimated reward function is straightforward: the weight vector at iteration i is transformed to an approximate set of gains via the relationship in Eq. 14. For a more complex controller model, this step is often replaced by a call to a reinforcement learning algorithm to recover a policy that optimizes the estimated reward function. Then, the parameters governing the apprenticeship learning procedure are also defined. In this particular application, the initial guess corresponds to a random policy, while the algorithm termination conditions include both $\epsilon < 1 \times 10^5$ and a maximum number of iterations based on computation time. Following formulation of the apprenticeship learning algorithm in the context of the spacecraft maneuver design example in this preliminary analysis, three scenarios are defined to explore both the accuracy of the results and the influence of the input data set on these results. In the first two cases, the expert policy trajectories begin near apolune along the NRHO and the LQR gains are selected to reflect either rendezvous and station-keeping behavior, i.e. placing an emphasis on minimizing state error or minimizing control usage, respectively. The last case is presented to recover maneuver design policies for trajectories beginning in the vicinity of a wider variety of states along the reference orbit. The same combinations of gains used in the first case is used in this third case to reflect rendezvous behavior. While the LQR cost function gains are constant, the relative dynamics influencing the maneuvering spacecraft will vary depending on the location of the spacecraft relative to the reference orbit. In this final case, apprenticeship learning is used to recover the controller defined by the LQR gains

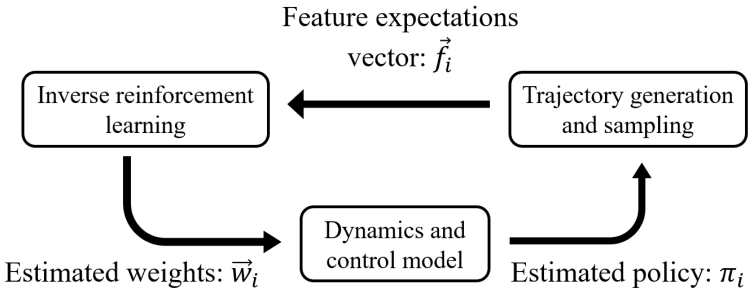


Fig. 4 Diagram of the apprenticeship learning process used to iteratively update an apprentice policy to closely match the performance of an expert policy.

using information from the entire reference orbit, significantly increasing the diversity of the expert demonstrations. These scenarios, labeled Case 1-3, are summarized in Table 1 along with the characteristics of the associated input data sets. In this table, the initial state distribution column lists the location/s along the NRHO used to generate the initial conditions: either at apoapsis, or distributed along the NRHO. The trajectory duration column lists the propagation time of each generated trajectory, defined as a fraction of the period, T, of the NRHO in the rotating frame; this time is also serves as the time horizon in the LQR control scheme. Finally, N is the number of trajectories generated from the expert and estimated policies, indicating the size of the expert data set.

Table 1 Summary of distinct maneuver design cases.

Case	Type of Control	Initial State Distribution	Trajectory Duration	N
1	Rendezvous	Apolune	$2/5\,T$	50
2	Station-Keeping	Apolune	$2/5\,T$	50
3	Rendezvous	Random, along entire orbit	$1/2\,T$	50

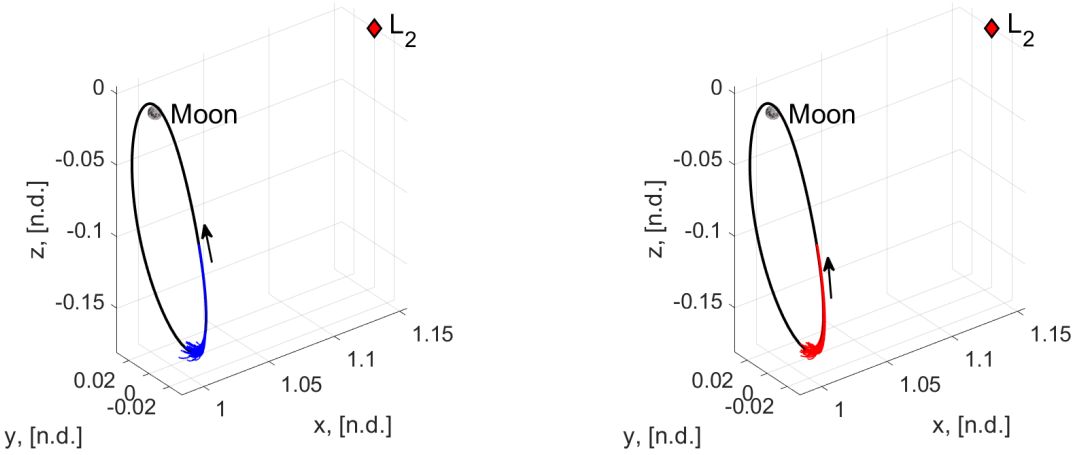
A. Case 1: Rendezvous from Apolune

In the first maneuver planning scenario, the LQR gains are selected consistent with a rendezvous scenario where position and velocity errors are most significantly penalized. Large gains of 10^5 and 10^2 are associated with position and velocity errors, respectively, while lower gains of 10 are associated with control use; the corresponding controller prioritizes a low state error with less emphasis on limiting propellant mass usage. The nine LQR gains selected for the expert policy to produce this behavior in Case 1 are listed in the "Expert Gains" column of Table 2. To evaluate each of the expert and apprentice policies, initial conditions are defined using a Gaussian distribution relative to the apolune location along the reference trajectory. The standard deviation of the initial position error is set equal to $\sigma_r = 1000$ km while the velocity error standard deviation is equal to $\sigma_v = 1$ m/s and the time horizon of the LQR controller is set to 2/5 of the orbit period of the NRHO in the rotating frame, i.e., approximately 2.5 days. Additionally, for both the expert and each iteration of the apprentice policy, 50 trajectories are used to generate the expert feature expectations vector based on the feature basis set defined in Eq. 12 and each trajectory is sampled to produce 1000 states distributed evenly in time. The components of the expert feature expectations vector, $\vec{f}_E$, are listed in the second column of Table 2.

The apprenticeship learning process is applied to recover an estimate of the expert's policy in Case 1 and, as a by-product, an estimate of the LQR gains. These estimated gains are used to generate the trajectories associated with the estimated or apprentice policy and, therefore, compute the apprentice feature expectations vector. The final estimated feature expectations vector and gains of the apprentice are listed in the third and fifth columns, respectively, of Table 2. Comparing the expert feature expectations vector, $\vec{f}_E$, and the final feature expectations vector from the apprentice policy, $\vec{f}$, the order of magnitude between each of the position, velocity and control components of the feature expectations are similar. Furthermore, as a by-product of the apprenticeship learning procedure, the estimated reward weights of the policy generated with apprenticeship learning are observed to follow the approximate structure of the known expert

Table 2 Summary of features expectations and LQR gains for the expert and apprentice policies in Case 1.

$\vec{\phi}$	Expert $\vec{f}_E$	Estimated $\vec{f}$	Expert Gains	Estimated Gains
δx^2	1.194e-03	1.428e-03	1.000e+05	1.961e+05
δy^2	1.260e-03	1.179e-03	1.000e+05	6.170e+05
δz^2	1.632e-03	4.341e-04	1.000e+05	8.300e+05
$\delta \dot{x}^2$	3.841e-02	6.376e-02	1.000e+02	1.376e+02
$\delta \dot{y}^2$	4.128e-02	1.648e-02	1.000e+02	2.495e+04
$\delta \dot{z}^2$	5.110e-02	2.478e-02	1.000e+02	3.355e+02
u_x^2	3.741e+00	9.719e+00	1.000e+01	1.000e+01
u_y^2	4.113e+00	2.039e+00	1.000e+01	4.170e+01
u_z^2	5.468e+00	4.756e+00	1.000e+01	2.592e+01



(a) Expert Trajectories **(b) Apprentice Trajectories**

Fig. 5 Case 1 trajectories generated from the expert and recovered apprentice policies for a rendezvous scenario.

weights: the estimated weights place higher emphasis on state error minimization, and specifically place a higher reward on minimizing position error than minimizing velocity error, while the lowest estimated weight is placed on minimizing control usage. However, as expected due to the nature of apprenticeship learning, these gains are not recovered exactly. Furthermore, the gain calculated from the $\delta\dot{y}^2$-component of the weight vector is significantly larger than the other two gains corresponding to velocity components. These differences between the original and recovered gains may also be due to nonuniqueness in the reward function, the specific termination conditions, relative scaling between the components of the feature expectations vector and the scaling strategy used to convert the weight vector into a set of LQR gains.

To compare the characteristics of the trajectories generated using the apprentice and expert policies, each of the state and control components are analyzed. Trajectories sampled from the expert policy and the estimated apprentice policy are plotted in the Earth-Moon rotating frame in Fig. 5, where expert trajectories are displayed in blue and apprentice trajectories are displayed in red. In this figure, the original reference NRHO is plotted in black while the L_2 equilibrium point is indicated as a red diamond. The expert and apprentice trajectories are observed to exhibit similar characteristics in quickly converging towards the NRHO. A more detailed comparison between the apprentice and expert trajectory is then achieved by analyzing the individual time histories of the state and control components: Figure 6 overlays the Euclidean norm of the position error for the expert and apprentice trajectories, Fig. 7 portrays the Euclidean norm of the velocity error for the expert and apprentice trajectories, and Fig. 8 overlays the Euclidean norm of the control usage for the expert and apprentice trajectories. In each figure, parameters associated with apprentice are plotted in red while those associated with the expert are depicted in blue. Using these figures as a reference, the time histories of the position and velocity magnitudes relative to the NRHO over the specified time horizon are on similar orders of magnitude and exhibit similar characteristics for both the apprentice and the expert. The magnitude of the control acceleration over time is also consistent in both order of magnitude and characteristics between the expert and apprentice. Of course, there are some minor deviations between the expert and apprentice in these three time history plots. However, these minor deviations exist only for a subset of the trajectories generated within the Gaussian distribution of initial conditions in the vicinity of the apolune location along the NRHO – and only for a portion of the specified time horizon. Thus, this analysis reveals that the policy recovered by apprenticeship learning is able to capture the expert behavior corresponding to the selected expert LQR cost function for this scenario reflecting rendezvous from apolune.

B. Case 2: Station-Keeping from Apolune

The second scenario uses LQR gains designed to generate station-keeping behavior, where the spacecraft trajectory simply remains bounded in the vicinity of the reference orbit. To achieve this type of behavior via the LQR controller,

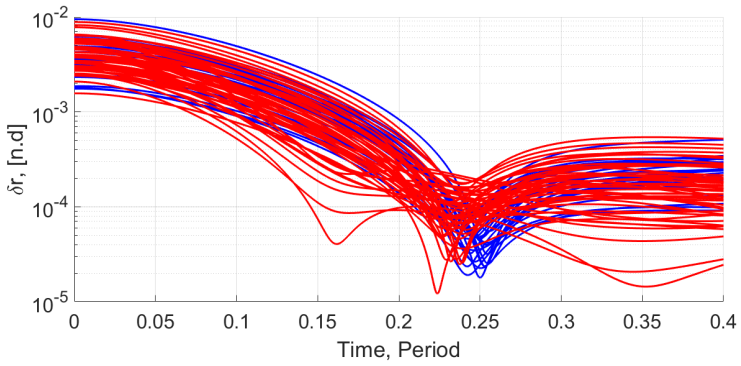


Fig. 6 Position error over time for trajectories generated from the expert policy (blue) and apprentice policy (red).

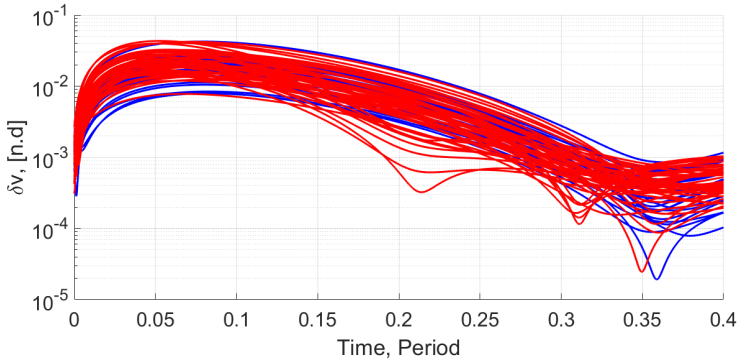


Fig. 7 Velocity error over time for trajectories generated from the expert policy (blue) and apprentice policy (red).

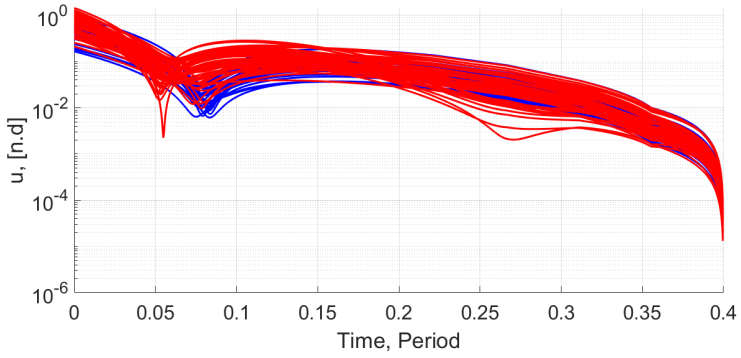
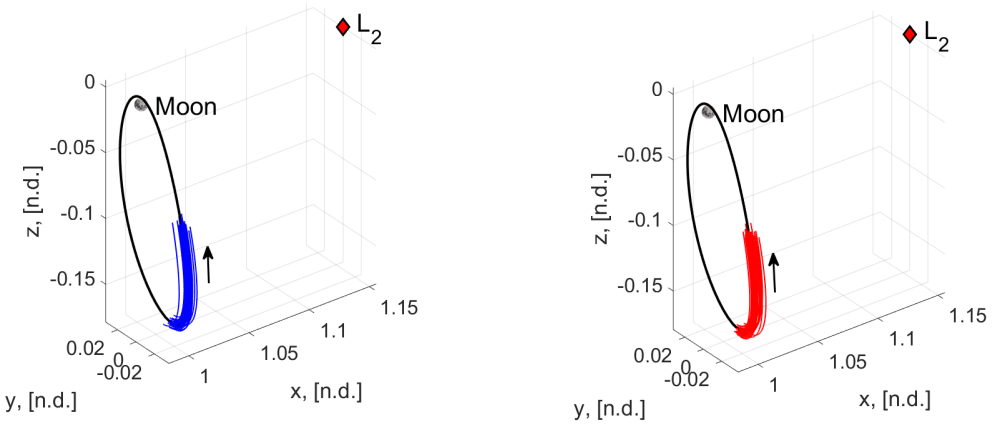


Fig. 8 Control usage over time for trajectories generated from the expert policy (blue) and apprentice policy (red).

the largest penalties – i.e., the largest gains – are associated with control usage, while lower gains are assigned to position and velocity errors. The exact LQR gains used to generate the expert trajectories are listed in the fourth column of Table 3. Trajectories generated using the expert policy for station-keeping via this LQR controller are plotted in blue in Fig. 9a in the Earth-Moon rotating frame. The initial conditions for these trajectories are defined within a Gaussian distribution relative to the apolune location along the NRHO using an initial position error standard deviation of σ_r = 1000 km and velocity error standard deviation of σ_v = 1 m/s. The simulation duration is set to 2.5 days and 50 trajectories are generated with 1000 states sampled evenly in time. After computing the expert feature expectations

Table 3 Case 2 summary of features expectations and LQR gains for the expert and apprentice policies.

$\vec{\phi}$	Expert $\vec{f}_E$	Estimated $\vec{f}$	Expert Gains	Estimated Gains
δx^2	6.799e-03	4.098e-03	1.000e+01	3.516e+02
δy^2	5.708e-03	3.043e-03	1.000e+01	3.259e+01
δz^2	5.699e-03	3.225e-03	1.000e+01	3.052e+01
$\delta \dot{x}^2$	2.224e-03	1.276e-03	1.000e+01	4.608e+03
$\delta \dot{y}^2$	8.321e-03	7.301e-03	1.000e+01	1.464e+03
$\delta \dot{z}^2$	1.636e-02	1.638e-02	1.000e+01	1.000e+01
δu_x^2	5.423e-12	1.489e-07	1.000e+05	1.095e+05
δu_y^2	5.086e-12	1.949e-07	1.000e+05	9.010e+04
δu_z^2	4.425e-11	8.966e-09	1.000e+05	6.726e+04



(a) Expert Trajectories

(b) Apprentice Trajectories

Fig. 9 Case 2 trajectories sampled from the expert and converged apprentice policy for a station-keeping example.

vector and applying the apprenticeship learning process, the recovered feature expectations vector of the apprentice and associated gains are listed in Table 3 for comparison to the expert. Analysis of these results reveals that the recovered apprentice feature expectations vector possesses values on the same order of magnitude as the apprentice for the position and velocity error feature basis functions. However, the order of magnitude of control components of the apprentice feature expectations vector differ significantly from those of the expert – yet, they are still several orders of magnitude less than the position and velocity components, consistent with the expert. Such a deviation in the apprentice feature expectations vector from the expert is likely due to the large variance in the order of magnitude of the components of the vector. This ill-conditioning between the components of the feature expectations vector motivates the exploration of feature scaling strategies in further analyses. Furthermore, analysis of the estimated gains reveals that the apprentice controller or reward function generally reflects the qualitative goals of the original controller: the control usage is penalized more heavily than the state error. However, there are significant deviations in the order of magnitude between the individual gains associated with position and velocity error in the LQR controller – inconsistent with the original LQR gains corresponding to the expert.

Further insight into the deviations between the expert and apprentice policies in Case 2 is gained through a comparison between each of the state and control components along the associated trajectory sets. In particular, trajectories generating using both the expert and apprentice policies are plotted in configuration space in the Earth-Moon system in Fig. 9; trajectories generated by the expert policy are plotted in blue and trajectories associated with the apprentice are displayed in red. The apprentice trajectories are observed to exhibit a similar general behavior to the

trajectories generated by the expert LQR controller; however, is it useful to analyze in more detail the time histories of the individual components of the trajectories and maneuver profiles. The norm of the position error, velocity error, and control usage over time of the trajectories associated with the expert (blue) and apprentice (red) policies are included in Fig. 10, Fig. 11, and Fig. 12 respectively. For each of these three quantities, the trajectories associated with the two policies are observed to have similar behavior – and distinctly different characteristics from the trajectories in Case 1. Thus, the apprenticeship learning algorithm successfully recovers the overall intention of the expert policies when the LQR controller gains are selected to produce specific trajectory behavior. However, there is a significant difference in the maneuver profile between the apprentice and expert, as displayed in Fig. 12. This difference is attributed to the inconsistent order of magnitudes of the feature expectations vector between the apprentice and expert. Recall that this difference that is likely due to ill-conditioning between the components of the feature expectations vector, warranting further analysis of feature scaling strategies.

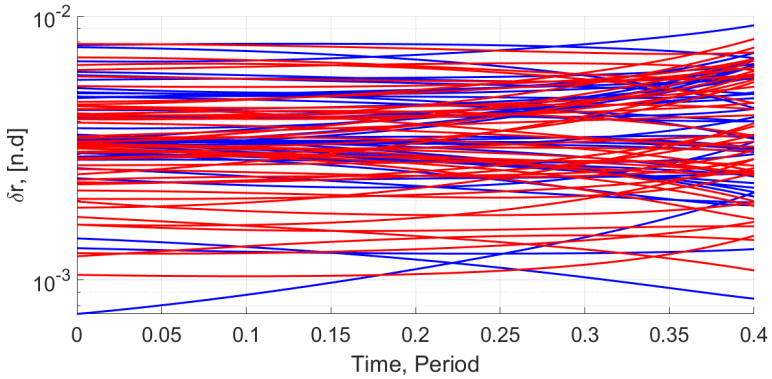


Fig. 10 Position error over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

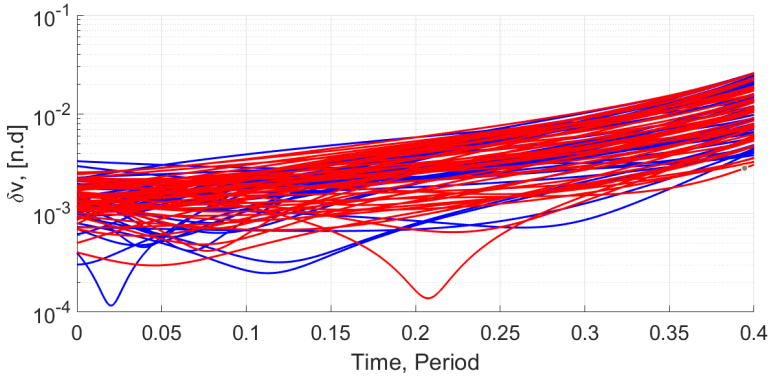


Fig. 11 Velocity error over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

C. Case 3

In case 3, apprenticeship learning is applied to an input data set composed of rendezvous trajectories associated with initial conditions that are perturbed from a variety of fixed points along the reference NRHO orbit. This scenario is used to evaluate the results of the apprenticeship learning when the input data set exhibits increased diversity. Since the NRHO exhibits a large range of relative distances to the Moon and, therefore, sensitivities in the relative motion dynamics, this increase in the diversity of the input data set is straightforwardly achieved by generating trajectories beginning near various locations along the reference orbit. To create the trajectories associated with the expert policy, the initial position error standard deviation is set at $\sigma_r = 100$ km and velocity error standard deviation of $\sigma_v = 1$ m/s.

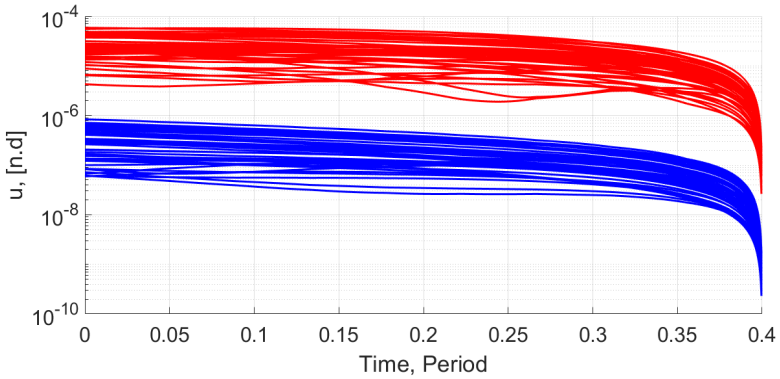


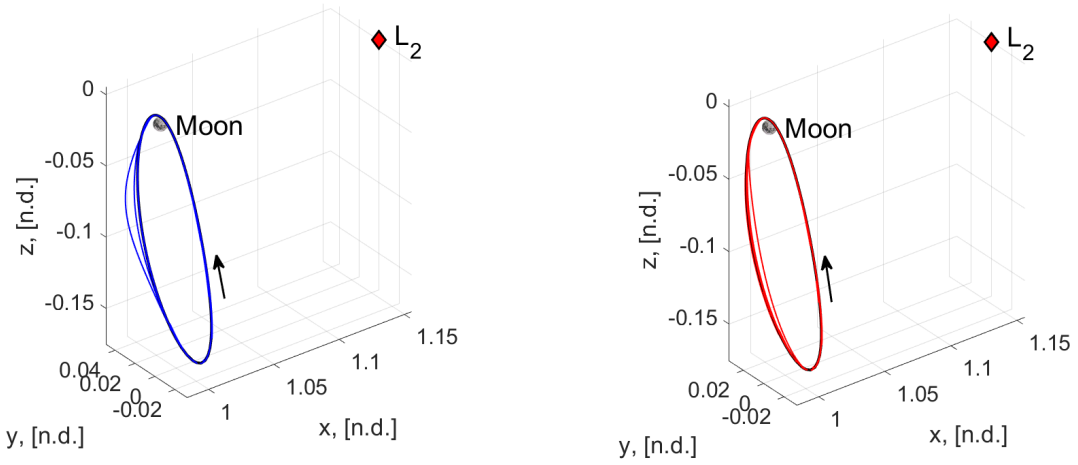
Fig. 12 Control usage over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

These standard deviations are used to define the Gaussian distribution relative to each randomly-selected fixed point along the NRHO. Each trajectory associated with an initial condition near the NRHO is propagated forward in time under the dynamics of the CR3BP for one half the period of the NRHO orbit, i.e approximately 3 days, with the same set of LQR gains. Using this procedure, 50 trajectories are generated from perturbations relative to fixed points along the entire orbit, except those that lie within $0.5T$ of perilune; each trajectory is sampled at 1000 evenly-spaced times. For this example, the same LQR gains from the rendezvous behavior in Case 1 are used, as listed in the fourth column of Table 4, to enable a clear comparison between the two cases. The 500 expert trajectories, initialized relative to random fixed points along the reference trajectory, are plotted in blue in the Earth-Moon rotating frame in Fig. 13a. The value of the expert weights, expert feature expectations, apprentice weights, and apprentice feature expectations for the third case are listed in Table 4 for comparison.

Comparing the feature expectations between the expert policy and the estimate policy, the two vector quantities are only similar to within an order of magnitude. Both feature expectations possess low position error feature expectations on a similar order of magnitude and velocity errors feature expectations also with similar orders of magnitude – although there is a larger difference between the exact value for the individual components than in Case 1 where the rendezvous trajectories are generated relative to a single fixed point along the NRHO. Notably, both the expert and estimated feature expectations exhibit the same characteristic of a smaller u_x^2 feature expectation than the u_y^2 and u_z^2 feature expectations. While the cause of the difference between control usage features is a product of the complex underlying dynamics and controller in this scenario, the recovery of the difference in these components of the feature expectations vector demonstrates the capability of the apprenticeship learning algorithm to recover a policy with a similar structure to that of the expert. For this case, the weights recovered by apprenticeship learning does not closely match the true

Table 4 Case 3 summary of features expectations and LQR gains for the expert and apprentice policies for trajectories sampled across the entire reference orbit.

$\vec{\phi}$	Expert $\vec{f}_E$	Estimated $\vec{f}$	Expert Gains	Estimated Gains
δx^2	4.963e-04	3.298e-04	1.000e+05	1.359e+07
δy^2	2.428e-03	2.359e-04	1.000e+05	2.779e+06
δz^2	4.866e-04	1.327e-04	1.000e+05	1.386e+07
$\delta \dot{x}^2$	6.813e-02	4.093e-02	1.000e+02	9.904e+04
$\delta \dot{y}^2$	2.621e-01	5.799e-02	1.000e+02	2.574e+04
$\delta \dot{z}^2$	1.258e-01	4.815e-02	1.000e+02	5.362e+04
δu_x^2	5.530e+00	3.502e+00	1.000e+01	1.220e+03
δu_y^2	6.747e+02	6.171e+02	1.000e+01	1.000e+01
δu_z^2	3.245e+02	4.476e+02	1.000e+01	2.080e+01



(a) Expert Trajectories

(b) Apprentice Trajectories

Fig. 13 Case 3 trajectories sampled from the expert and converged apprentice policy for a rendezvous example.

weights specified a priori by the expert LQR cost function. However, recovering the exact weights of the expert reward function is not guaranteed by the algorithm – thus, this result is not unexpected. Furthermore, while the reward function associated with the LQR controller is constant for trajectories beginning near any fixed point along the NRHO, a close match between the expert and apprentice feature expectations vectors and, therefore, policies may not be feasible when the input data set is too diverse.

Further insight into the deviations between the expert and apprentice policies in Case 3 is gained through a comparison between each of the state and control components along the associated trajectory sets. Trajectories generated from both the expert (blue) and apprentice (red) policies are compared in Fig. 13 in the configuration space of the Earth-Moon rotating frame. As opposed to case 1 and 2, the sampled trajectories for the expert and estimated policy are generated near fixed points around the entire reference orbit. A notable characteristic of the behavior of the expert policy trajectories is the trajectories that are initialized closer to lunar periapsis admit a large visible position error before converging to the reference orbit; a similar characteristic may be observed in the trajectories generated from the apprentice policy. However, a more detailed analysis in the deviation between these trajectories is achieved using the individual time histories of quantities along each trajectory set. Specifically, the norm of the position error, velocity error, and control usage over time for trajectories generated via the expert (blue) and apprentice (red) policies are displayed in Fig. 14, Fig. 15, and Fig. 16, respectively. Overall, the characteristics of the trajectories generated via both the expert and apprentice policies are summarized by a large initial control usage that decreases in time, as well as gradually decreasing position and velocity error over time. However, while the general trends and orders of magnitude in these quantities are similar between the expert and apprentice trajectories, there are significant differences in the time histories of position and velocity error and control usage. For instance, in Fig. 15, the trajectories generated via the apprentice policy exhibit small oscillations in the magnitude of the velocity error – these characteristics are not consistent with the shape of the magnitude of the velocity error evaluated along the expert trajectories. This local mismatch between the trajectories may be due to the deviations between the $\delta\dot{y}^2$ and $\delta\dot{z}^2$ components of the feature expectations vector for both the expert and apprentice. Further exploration into whether these deviations may be decreased is warranted: either through feature scaling, modification of the termination conditions or updates to the value of α used to transform the weight vector to the LQR gain set during each iteration of the apprenticeship learning algorithm.

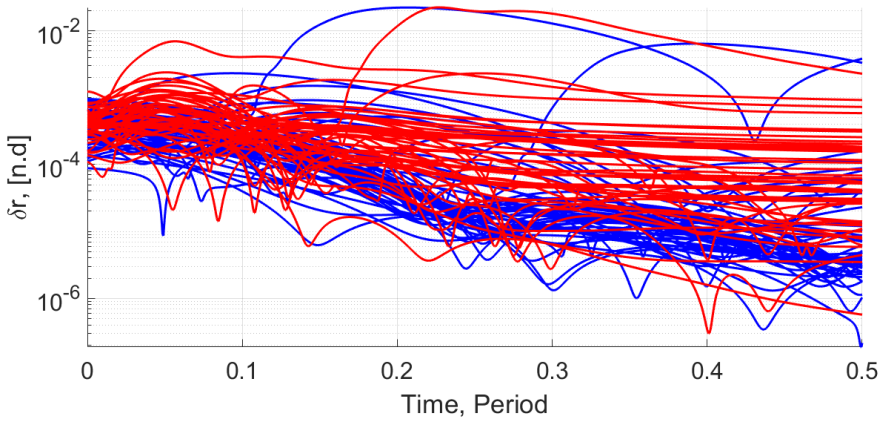


Fig. 14 Position error over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

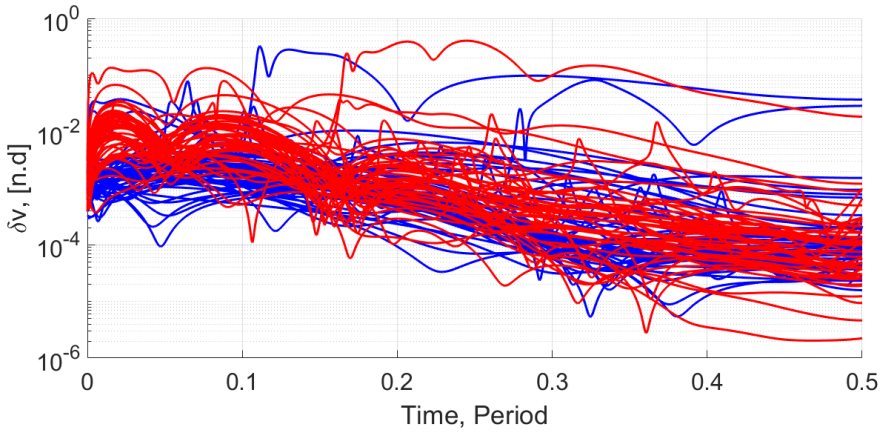


Fig. 15 Velocity error over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

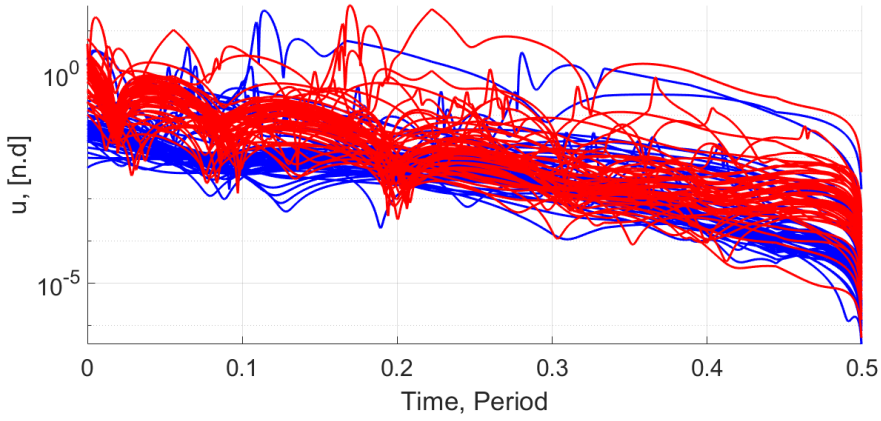


Fig. 16 Control usage over time for trajectories sampled from the expert policy (blue) and sampled from the apprentice policy (red).

IV. Conclusion

Apprenticeship learning, via inforcement reinforcement learning, is used to recover the strategy of a maneuver planner for a spacecraft operating in the multi-body gravitational environment of cislunar space. In this preliminary analysis, this procedure is implemented and explored to recover maneuvers defined by an LQR controller. Specifically, the LQR controller is applied to rendezvous and station-keeping scenarios for a spacecraft operating near an Earth-Moon L_2 NRHO reference orbit. Two cases of expert LQR cost function gains are selected to evaluate the utility of the apprenticeship learning algorithm in recreating distinctly different behaviors for controlled trajectories relative to the NRHO. The results of this algorithm are evaluated for both rendezvous and station-keeping examples for trajectories beginning near a single initial condition at apolune on the reference trajectory. In the rendezvous scenario, the policy recovered by apprenticeship learning exhibits closely matching feature expectations to the expert policy input to the algorithm. These expert and apprentice policies are compared by analyzing the trajectories generated from both policies in the Earth-Moon rotating frame, as well as comparing the Euclidean norm of the position error, velocity error, and control usage over time. In the rendezvous scenario, the apprentice policy successfully mimics the characteristics of the expert. While the apprenticeship learning algorithm does not claim to recover the true reward function of the expert, the reward weights estimated by the algorithm after normalization are observed to be similar to the weights of the expert reward function. In the station-keeping scenario, the apprenticeship learning algorithm implemented in this analysis produces an estimated policy that recreates the general structure of the expert policy. However, this apprentice policy does not sufficiently capture the control usage behavior associated with the expert of the policy. This mismatch is likely due to ill-conditioning between the components of the feature expectation vectors, thereby motivating further analysis into feature scaling strategies. Next, the rendezvous scenario is modified by introducing a more diverse set of expert demonstrations: trajectories generated via the expert policy are initialized relative to a variety of fixed points along the NRHO. In this case, the general structure of the LQR controller is approximately recovered and the trajectories generated by the apprentice policy admit position, velocity and control usage errors that are on the same order of magnitude as those generated via the expert policy. However, variations between the trajectories generated by the apprentice and expert policies occur on smaller time scales across the specified time horizon. Such deviations may occur due to a combination of ill-conditioning between the components of the feature expectations vector and the strategy for transforming between the weight vector recovered after the inverse reinforcement learning step and estimated LQR controller gains. However, it is more likely that diversity of the input data set negatively impacts the recovery of an apprentice policy that is simultaneously accurate and generalizable for trajectories beginning near various locations along the NRHO. Nevertheless, this analysis presents a preliminary step towards using techniques derived from machine learning to support the summarization of maneuver planning strategies for a spacecraft operating in the complex gravitational environment of cislunar space. Such techniques may be particularly valuable for scenarios reflecting more complex control schemes where an exact analytical expression cannot summarize the maneuvers designed by a large-scale optimization or a human flight dynamicist.

Acknowledgments

This work was supported by an Early Stage Innovations grant from NASA's Space Technology Research Grants Program, under NASA grant 80NSSC19K0222.

References

[1] Levine, S., Popovic, Z., and Koltun, V., "Nonlinear Inverse Reinforcement Learning with Gaussian Processes," *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., 2011, pp. 19–27.

[2] Dvijotham, K., and Todorov, E., "Inverse Optimal Control with Linearly-Solvable MDPs," *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress, USA, 2010, pp. 335–342.

[3] Ng, A. Y., and Russell, S. J., "Algorithms for Inverse Reinforcement Learning," *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 663–670.

[4] Abbeel, P., and Ng, A. Y., "Apprenticeship Learning via Inverse Reinforcement Learning," *Proceedings of the Twenty-first International Conference on Machine Learning*, ACM, New York, NY, USA, 2004, pp. 1–. doi:10.1145/1015330.1015430.

[5] Arora, S., and Doshi, P., "A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress," Ph.D. thesis, 2018.

[6] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A., "Maximum Margin Planning," *Proceedings of the 23rd International Conference on Machine Learning*, ACM, New York, NY, USA, 2006, pp. 729–736. doi:10.1145/1143844.1143936.

[7] Linares, R., and Furfaro, R., "Space Objects Maneuvering Detection and Prediction via Inverse Reinforcement Learning," 2017.

[8] Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K., "Maximum Entropy Inverse Reinforcement Learning," *AAAI Conference on Artificial Intelligence*, 2008, pp. 1433–1438.

[9] Fu, J., Luo, K., and Levine, S., "Learning Robust Rewards with Adversarial Inverse Reinforcement Learning," *CoRR*, Vol. abs/1710.11248, 2017. URL http://arxiv.org/abs/1710.11248.

[10] Doerr, B., Linares, R., and Furfaro, R., "Space Objects Maneuvering Prediction via Maximum Causal Entropy Inverse Reinforcement Learning," *arXiv preprint arXiv:1911.00489*, 2019.

[11] Zhifei, S., and Joo, E., "A Survey of Inverse Reinforcement Learning Techniques," *International Journal of Intelligent Computing and Cybernetics*, Vol. 5, 2012, pp. 293–311. doi:10.1108/17563781211255862.

[12] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y., "An Application of Reinforcement Learning to Aerobatic Helicopter Flight," *Advances in Neural Information Processing Systems*, 2007, pp. 1–8. doi:10.7551/mitpress/7503.003.0006.

[13] Abbeel, P., Coates, A., and Ng, A. Y., "Autonomous Helicopter Aerobatics Through Apprenticeship Learning," *International Journal of Robotics Research*, Vol. 29, No. 13, 2010, pp. 1608–1639. doi:10.1177/0278364910371999.

[14] Whitley, R., and Martinez, R., "Options for staging orbits in cislunar space," *2016 IEEE Aerospace Conference*, 2016, pp. 1–9. doi:10.1109/AERO.2016.7500635.

[15] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., *"Dynamical Systems, the Three-Body Problem and Space Mission Design"*, 2011. Marsden Books.

[16] Lee, D. E., "White Paper: Gateway Destination Orbit Model: A Continuous 15 year NRHO Reference Trajectory," 2019.