Autonomous Trajectory Planning for Cislunar Space

Brian McCarthy
Advanced Mission Design Engineer, a.i. solutions, Inc
Fabio Chiappina
Software Systems Engineer, a.i. solutions, Inc

ABSTRACT

As civil and defense infrastructure in cislunar space continues to expand, effective trajectory planning techniques will prove a critical component to cislunar SDA. Responding to possible threats by quickly deploying assets in the cislunar regime requires a method for rapid delivery of multiple viable trajectory options to decision makers. This investigation demonstrates a process to rapidly generate candidate trajectories to move through cislunar space using clustering and other machine learning techniques. Several examples of autonomously constructed transfers between a variety of periodic orbits are presented. The investigation focuses on reducing user input to develop the trajectories, where traditional methods would have required time-intensive analysis by industry experts. The sensitive cislunar dynamics are a new paradigm compared to near-Earth dynamics, and this investigation attempts to tackle the challenges of effectively assessing and producing viable paths through this space using novel techniques.

1. INTRODUCTION

With the proliferation of civil and defense infrastructure in cislunar space, effective methods to rapidly plan trajectories throughout the regime are becoming increasingly necessary. However, given that cislunar dynamics are inherently chaotic and highly sensitive, finding a viable path through space that connects any two points is challenging. The potential types of trajectory geometries that exist in cislunar space are widely varied: conic sections, which characterize near-Earth orbits, do not approximate the paths that a vehicle can take through cislunar space. Different models can be employed to assess potential paths in cislunar space, but employing these effectively requires specialized domain expertise in multi-body dynamics.

In this investigation, an approach is discussed for finding viable paths starting in a simplified dynamical model, which is used to provide initial guesses to a higher-fidelity model, demonstrating an autonomous end-to-end trajectory design process. Ultimately, an effective, streamlined end-to-end trajectory construction process enables rapid delivery of viable options to decision makers, improving threat assessment and allowing quicker deployment of assets to respond to possible threats. This process could also be adapted to predict where and how rapidly adversarial assets could move in cislunar space - an increasingly critical capability from the perspective of space domain awareness.

Obtaining an initial guess trajectory through cislunar space by traditional methods is a time-intensive process, requiring a high level of astrodynamics expertise, specifically in multi-body dynamical structures. An initial guess must then be provided to a targeting or optimization tool to produce an end-to-end design, and that initial guess typically drives which basin of convergence the optimized solution tends toward. The initial guess therefore often drives the overall time of flight and maneuver cost of a cislunar mission. To reduce dependency on constant, high-level astrodynamics expertise, an automated trajectory planning process is developed to rapidly construct initial guess cislunar trajectories, leveraging various known dynamical structures in the Earth-Moon Circular Restricted Three-Body Problem (CR3BP).

Several previous investigations explored the initial guess problem in the CR3BP using trajectory classification. Smith and Bosanac developed an approach to leverage motion primitives and multi-objective collocation to build end-to-end trajectories in the Earth-Moon CR3BP [15]. Additionally, Miceli et al. leverages a similar framework to design trajectories in the Neptune-Triton system [12, 11]. Similarly, Das et al. explored using automated path planning algorithms for various low-thrust trajectory design in multi-body systems [4].

This investigation leverages the techniques developed by Smith and Bosanac using motion primitives in concert with dimensionality reduction algorithms to improve clustering performance, as well as a novel convolutional neural network approach for extracting visually-based trajectory feature vectors to improve the logical grouping of visually similar trajectory solutions. Additionally, this investigation uses an open source database of periodic orbits to build the motion primitive search space [6]. Lastly, a transition process using the initial guess to a higher-fidelity ephemeris model is performed to construct an end-to-end trajectory using a low-thrust engine. A similar process was developed by Smith and Bosanac for high-thrust engines; thus, this analysis demonstrates the viability of this approach for low-thrust engine applications, similar to Das et al. The algorithm is illustrated at a high-level in Figure 1; each illustrated stage in the process is discussed in greater detail below.

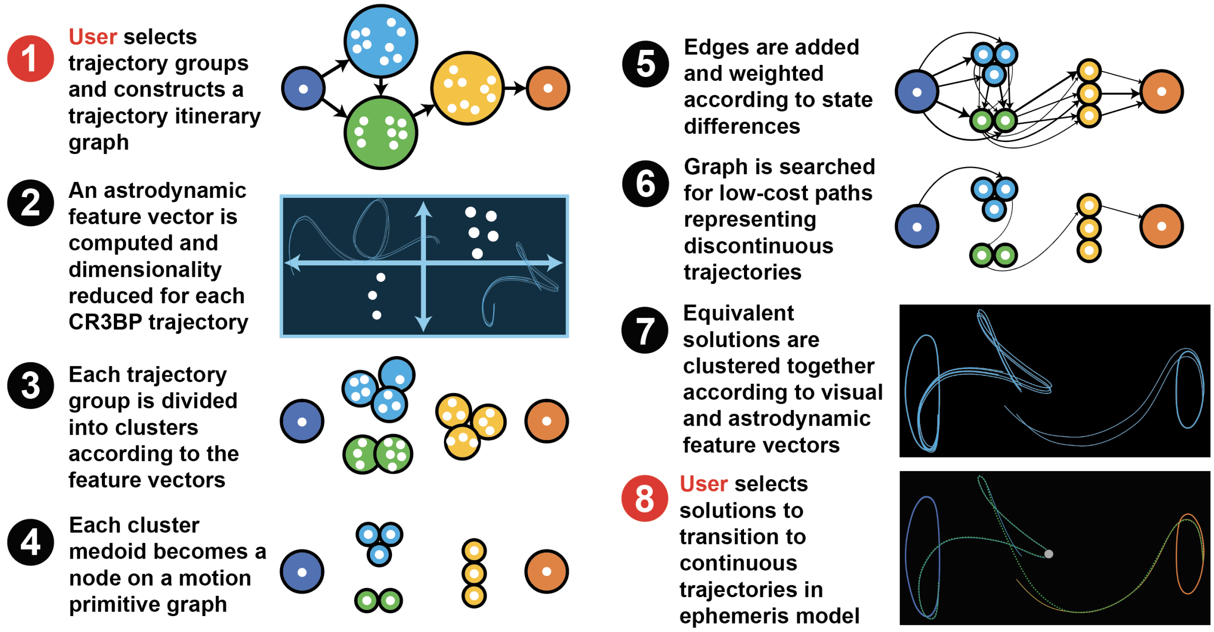


Fig. 1: A high-level illustration of the algorithm used by the autonomous path planning tool. User inputs are highlighted in red in the first and final steps; the remaining steps are performed autonomously.

2. DYNAMICAL MODELS

Two dynamical models are leveraged in this investigation: the Circular Restricted Three-Body Problem (CR3BP), used in initial guess construction, and a Sun-Earth-Moon ephemeris model, used to transition initial guesses into higher-fidelity trajectory solutions via FreeFlyer, a commercial astrodynamics software. The CR3BP is a more accurate and relevant low-fidelity model when operating in cislunar space in comparison to the two-body model. In this model, two gravitational bodies, denoted P_1 and P_2, remain in circular Keplerian orbits about their mutual barycenter (i.e., center of mass). A third body, typically a man-made satellite, P_3, moves under the gravitational influence of the two larger bodies and is assumed to be massless. The model is defined relative to a rotating coordinate system, where the $+\hat{x}$ direction is defined from the barycenter toward P_2. The $+\hat{z}$ direction is defined parallel to the direction of the orbital angular momentum vector for P_1 and P_2; the $\hat{y}$ direction completes the orthonormal triad. The position and velocity of P_3 relative to the barycenter in the rotating frame are defined as $\vec{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$, where the first three and the last three elements are the position and relative velocity components, respectively. The equations of motion for a particle moving in the CR3BP are a set of three, second-order scalar differential equations of motion:

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

The pseudo-potential is a scalar defined solely as a function of position and the CR3BP mass parameter, $\mu = M_2/(M_1 + M_2)$, where M_1 and M_2 are the masses of P_1 and P_2, respectively [17]. The pseudo-potential function takes the following form,

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

where $d = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ represent the distances of P_3 relative to P_1 and P_2, respectively. The CR3BP admits a single integral of the motion, commonly denoted the Jacobi Constant (JC). The Jacobi Constant is a function of the pseudo-potential and the relative velocity magnitude expressed in the rotating reference frame,

$$JC = 2U^* - v^2 \tag{3}$$

where $v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$. The Jacobi Constant is an energy-like quantity that characterizes motion in a CR3BP system and remains constant for all time over any ballistic arc propagated in the CR3BP. The CR3BP is a good approximation for a multi-body environment and the trajectory characteristics generally persist when transitioning results to a higher-fidelity ephemeris model.

3. INITIAL GUESS GENERATION

The autonomous trajectory planning process leverages several steps to produce end-to-end solutions. Much of this process follows the steps summarized by Smith and Bosanac [15] as well as Smith [14], with a few modifications for computational efficiency and further qualitative reduction of the solution space. The steps in the process are summarized here for completeness, highlighting where the modifications exist. A prerequisite step before the algorithm can be utilized is to collect a set of dynamical structures that leverage the natural dynamics of the Earth-Moon CR3BP. These dynamical structures include periodic orbits and associated stable and unstable manifold trajectories. An open-source database of periodic orbits is used to populate the set used in this investigation [6]. Of the periodic orbits that are unstable in the database, stable and unstable manifold trajectories were constructed to augment the set of dynamical structures. Examples of some of the periodic orbits considered from the database are rendered in Figure 2 in the Deep Space Trajectory Explorer (DSTE). The stable and unstable manifolds associated with these periodic orbits can serve as maneuver-free paths into and out of their originating orbit, making them excellent candidates to aid in construction of paths through cislunar space. Additionally, sections or partial arcs associated with periodic orbits can be used as building blocks to construct intermediate arcs along a path.

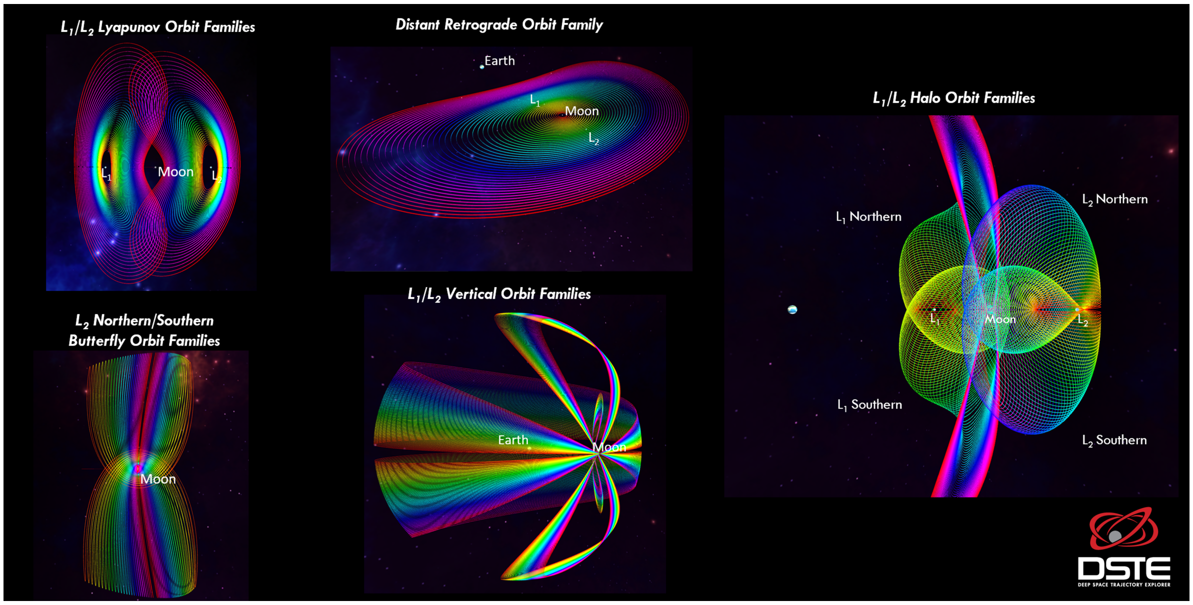


Fig. 2: Examples of orbit families in the Earth-Moon CR3BP rendered in the Deep Space Trajectory Explorer.

With the database of periodic orbits and manifold trajectories providing an expansive library of known trajectory paths in cislunar space, the user begins by selecting one or more departure and destination orbit(s) to define the desired start and end location(s) of the cislunar path. The ability to search for paths from/to a set of multiple departing/destination orbits is a novel capability demonstrated by this analysis and allows the user additional flexibility for designing trajectories in applications with a wide range of acceptable trajectory outcomes. The user also selects any number of additional periodic orbits and/or manifold trajectories from the database to serve as a set of intermediate trajectory arcs, segments of which can be combined to form a complex and low-cost path connecting the departing and destination orbit(s). The departing and destination orbits, along with each selected subset of intermediate trajectory arcs, are each represented as nodes on a high-level trajectory itinerary graph structure, as noted in step 1 of Figure 1. The user can permit the algorithm to search for connections between two sets of trajectories by adding to the trajectory itinerary graph an edge between the two nodes corresponding to those trajectory sets; an illustration of a trajectory itinerary graph and the implications associated with connecting two of its nodes with an edge is provided in Figure 3. Thus, a user experienced in trajectory design can carefully select sets of trajectories that they believe may be useful as intermediate arcs in the desired transfer, connecting via edges in the trajectory itinerary graph sets of trajectories that are likely to come close to intersecting in position-velocity space. A less experienced user can still create such a trajectory itinerary graph, but they may include a larger set of intermediate trajectories and/or use a denser (or even complete) set of edges; though viable, such an approach comes at the cost of slowing down the algorithm with extraneous trajectories unlikely to be useful in the desired transfer. An example of such a trajectory itinerary graph is presented in Figure 4, along with a visualization of the trajectories corresponding to the graph's nodes.

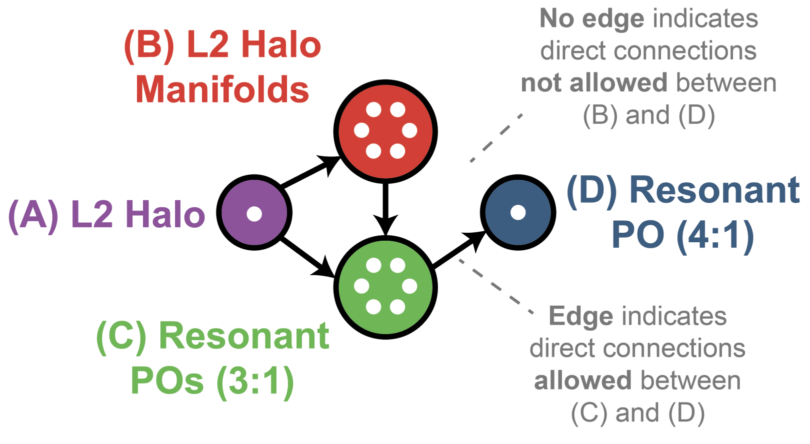


Fig. 3: An example high-level trajectory itinerary graph for a transfer between an L_2 Halo orbit and a $4:1$ resonant orbit highlighting the user's design decisions on whether to connect graph nodes with edges.

Next, an astrodynamic feature vector is computed encoding the dynamical properties of each of the selected cislunar orbits and manifold trajectories in the high-level trajectory itinerary graph, as illustrated in step 2 of Figure 1. These vectors are formed by concatenating position and velocity vectors at a predefined number of discretized states along the trajectory path. For periodic orbits, their Jacobi Constant and stability indices are also concatenated onto their feature vectors, whereas for manifold trajectories, the times between consecutive discretized states are included in their feature vectors. The representation of each trajectory as an astrodynamic feature vector allows each set of trajectories (each node in the high-level trajectory itinerary graph) to be divided via a clustering algorithm (discussed below) into groups of trajectories with similar astrodynamic properties.

Astrodynamic feature vectors fed into the clustering algorithm are high-dimensional; in this analysis, periodic orbit feature vectors contained 63 elements (ten discretized states, Jacobi Constant, and two stability indices), and manifold trajectory feature vectors contained 69 elements (ten discretized states and nine times between consecutive states). Since many clustering algorithms rely on computing the distances between vectors, such algorithms are prone to what is often referred to as the "curse of dimensionality", a phenomenon whereby the large volume of a higher-dimensional feature space is covered only quite sparsely by the dataset, which tends to reduce the effectiveness of algorithms that use distances between points in that feature space as a measure of similarity [1, 2]. Dimensionality reduction has been shown to improve similarity measurements and, consequently, the effectiveness of clustering algorithms for certain datasets [1, 16]. Two dimensionality reduction techniques were thus examined to make the clustering algorithm more effective: principal component analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP). PCA, which attempts to reduce the number of original variables into a set that maximally describes the variance [19], was examined first. But since this method is generally useful for sets of data where non-linear effects are minimal, an

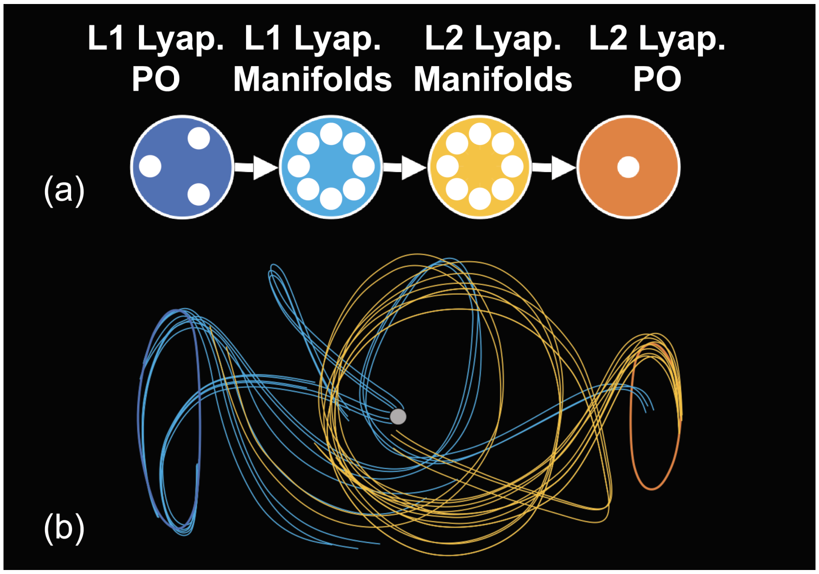


Fig. 4: (a) An example high-level trajectory itinerary graph for a transfer between an L_1 Lyapunov orbit and an L_2 Lyapunov orbit with four nodes, each containing one or more trajectories. (b) An illustration of the four sets of trajectories corresponding to the four nodes in the above trajectory itinerary graph.

alternative dimensionality reduction algorithm was also examined: UMAP, which attempts to capture more of the non-linear features of a given system [10]. However, the elements of the resulting reduced-dimension feature vectors that are produced by UMAP have a less interpretable physical meaning than those produced by PCA, which are tied to the feature dimensions of the original dataset along which the maximal variance is retained. Further analysis is necessary to determine which of these algorithms yields superior results for this application.

Once astrodynamic feature vectors have been computed and dimensionality-reduced, these vectors are clustered into sets of motion primitives using Weighted Evidence Accumulation Clustering (WEAC) [5]. WEAC relies on computing clustering assignments by running several clustering algorithms and aggregating the results, considering how commonly two points were grouped in the same cluster among all the returned clustering assignments. This allows WEAC to take advantage of the strengths of many different clustering algorithms, some of which may only be effective at finding clusters of a particular shape or with other specific qualities. In this analysis, WEAC used three popular clustering algorithms to produce intermediate clustering assignments: k-means, DBSCAN, and agglomerative clustering with Ward linkage. After each set of trajectories in the high-level trajectory itinerary graph has been grouped into clusters by WEAC, the medoid of each cluster (i.e., the element that is most similar to other elements in that same cluster) is extracted, as illustrated in steps 3 and 4 of Figure 1. Each of these medoids is considered a motion primitive - a dynamical building block that can be combined with other primitives to express more complex dynamical paths. The clustering process serves to reduce the large library of available trajectories to a smaller set that is still representative of the possible fundamental types of motion in the cislunar regime. A simplified illustration of the clustering procedure, the low-dimensional feature space onto which astrodynamic feature vectors are projected, and the extraction of cluster medoids is provided in Figure 5. Though this analysis used a three dimensional feature space, a two-dimensional space is illustrated for simplicity. Similarly, while the illustration depicts three relatively distinct trajectory sets, in practice, trajectories clustered by this analysis are unlikely to be so cleanly separable into clusters, and the feature space is likely to be more uniformly populated.

As shown in steps 4 and 5 of Figure 1, each motion primitive is represented as a node on a weighted, directed, motion primitive graph, where edge weights between two nodes encode the state differences between the two corresponding trajectory arcs. Each path through this motion primitive graph originating at the departing orbit's (or orbits') corresponding node(s) and terminating at the destination orbit's (or orbits') corresponding node(s) represents a discontinuous sequence of trajectory arcs that can be refined and transitioned into a continuous trajectory in a higher-fidelity dynamical model with the addition of propulsive maneuvers. Since the motion primitive graph edge weights represent state differences in position and velocity space, paths through the graph with lower total edge weight have smaller

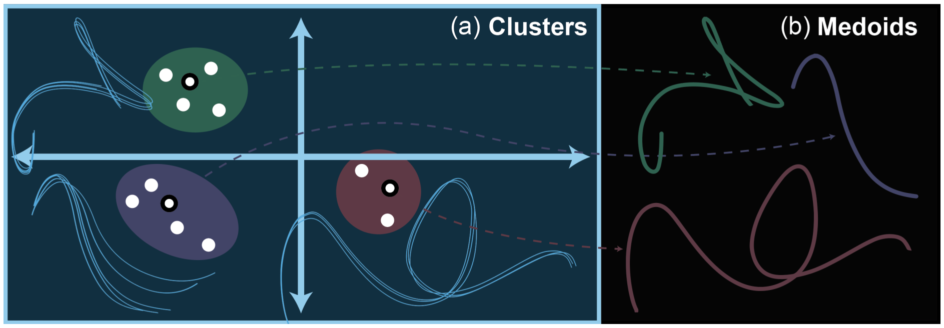


Fig. 5: (a) An illustration of the low-dimensional feature space onto which the astrodynamic feature vector – represented here as a white dot – of each trajectory is projected. Dots with a black outline represent the medoid, or representative element, of each cluster. In this simplified example, the clustering procedure groups these vectors into three clear clusters, each of which is depicted alongside its corresponding trajectories. (b) An illustration of the smaller trajectory set returned by the clustering procedure, consisting only of trajectories corresponding to the medoids of the clusters in (a).

total state discontinuities, meaning connecting the corresponding discontinuous trajectory arcs will require a lesser total propellant cost. A diagram of the graph traversal problem is illustrated in Figure 6, where the circle on the left side of the diagram represents a departing orbit node and the circle on the right represents a destination orbit node. The circles and directed arrows in the middle represent the intermediate nodes and the edges of the graph, where the thickness of the lines represent the edge weights of the corresponding edges. The lowest cost path through the graph is highlighted in pink, but the graph may contain any number of intermediate nodes connected by any arbitrary graph structure, and it can also utilize a set of nodes in place of a single departing or destination node.

With the motion primitive graph constructed, the graph is then searched for not just the single lowest-cost path from the departing orbit node(s) to the destination orbit node(s), but for a set of many different low-cost paths, thus providing the user with a set of many unique sequences of trajectory arcs for further refinement. The graph search is illustrated in step 6 of Figure 1. In a similar manner to Miceli et al., Yen's *k*-shortest paths algorithm, coupled with Dijkstra's algorithm to find each individual shortest path, is used as the search algorithm to efficiently construct the top k shortest paths from the departing orbit node(s) to the destination orbit node(s) [12]. The aforementioned process of the user down-sampling the large database of trajectories through their selection of the high-level trajectory itinerary graph, as well as automatically down-sampling again through the clustering algorithm and motion primitive extraction, results in two advantages during the motion primitive graph search: (1) the resulting graph is smaller, thus requiring less time to search, and (2) each path through the graph corresponds to a relatively distinct trajectory, where otherwise many of the lowest-cost paths would likely correspond to qualitatively similar trajectory sequences.

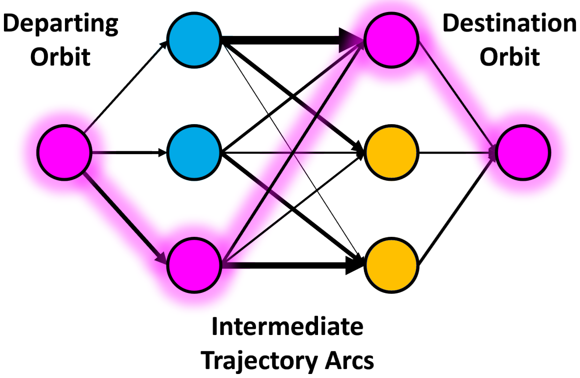


Fig. 6: An example motion primitive graph and returned low-cost path. The thickness of lines connecting any two nodes indicates the weight of that edge, and the highlighted path represents the lowest-cost path through the graph.

Once paths are constructed using motion primitive sets, the results from the previous step are further culled by grouping together trajectories with similar qualitative features, as shown in step 7 of Figure 1. Of the trajectories produced by the graph search process, despite efforts to limit the redundancy of returned solutions by winnowing out similar trajectories through the motion primitive clustering process, some of the returned trajectory paths may still be sufficiently similar that they would ultimately fall into the same convergence basin if passed to the higher-fidelity targeting/optimization process. Examples of three trajectory itineraries that possess similar qualitative properties are rendered on the left side of Figure 7. To avoid presenting each of these similar trajectories to the user as separate, unique solutions, a secondary clustering process is invoked to group together trajectory sequences with similar properties, further eliminating redundant solutions. Yet even for a set of trajectory sequences that are clearly qualitatively similar, their corresponding high-dimensional astrodynamic feature vectors and UMAP- or PCA-reduced-dimension equivalents may not necessarily remain close together in their respective feature spaces. Thus, when clustered according to astrodynamic feature vectors, qualitatively similar trajectories are not necessarily considered part of the same cluster, as shown on the right side of Figure 7.

A novel approach for remedying this qualitative similarity discrepancy in trajectory sequence clustering revolves around feature vectors that encode visual properties instead of astrodynamic properties. This approach leverages the Convolutional Neural Network (CNN), a powerful machine learning model that can be trained on a set of images to extract their semantic content and make predictions upon that content. CNN architectures include a series of convolutional layers, each of which serves a particular function in encoding the semantic content of the images upon which the network is trained, followed by some number of fully-connected layers. The set of convolutional layers encodes an input image array into a high-dimensional, single-column feature vector. Fully-connected layers learn to translate the convolutionally-computed feature vector into the desired output - perhaps an image class (e.g., the name of the object or action in the image), or an estimate of a continuous value (e.g., the number of people in an image of a crowd). Using a technique known as transfer learning, a CNN trained on one set of images can be used to make predictions on another set of images, requiring only a small number of training images, or even none at all. Typically, the earlier layers of the CNN tend to learn to respond to fundamental visual concepts like edges and basic shapes, whereas later layers tend to respond to increasingly abstract concepts like actions, objects, and people; transfer learning works by detaching the latest (most abstract and thus domain-specific) layers of a CNN, leaving behind pre-trained early layers that have already learned to extract basic visual concepts from images. In some applications, additional untrained layers are appended to these pre-trained layers, and these new layers are retrained on the new dataset of images to solve a new problem. In this investigation, however, no new layers are appended; instead, the convolutional layers of a CNN pre-trained as an image classifier are used as a generic image feature extractor - a network that translates a 3D input image array into a high-dimensional, single-column feature vector that encodes some of the visual properties of that image.

For each trajectory sequence found by the graph search algorithm, an image is constructed by layering image arrays of three projections of that trajectory sequence into the *xy*, *xz*, and *yz* planes in the Earth-Moon rotating frame, with each plane mapping to one of the three standard image color channels (R, G, and B). That image, an example of which is provided in Figure 10(b) below, is then fed through the pre-trained CNN feature extractor to produce a visual feature vector – a vector that encodes for its corresponding trajectory not its dynamical properties, but its visual properties. Using these visual feature vectors either instead of or in addition to the aforementioned astrodynamic feature vectors may yield more qualitatively intuitive trajectory sequence clustering results in that visually similar trajectories may be more likely to be grouped into the same cluster. This feature vector extraction process is summarized at a high-level by Figure 8. The CNN that is leveraged in this investigation is VGG-16, originally trained as a 1000-class image classifier on the ImageNet dataset [13]; the network's thirteen convolutional layers form the feature extractor, with all three of the fully-connection layers dropped. An illustration of the VGG-16 CNN architecture is shown in Figure 9, where the network is used to classify the input image as belonging to the class "boat".

Figure 10(a) shows an example of a sequence of trajectories returned by the motion primitive graph search process and plotted in the Earth-Moon rotating frame. Figure 10(b) shows the image corresponding to that same trajectory provided to the CNN feature extractor for classification. Note that there are three images stacked in Figure 10(b) that show different projections of the trajectory. The coloring corresponds a particular projection: red corresponds to the *xy* projection, blue corresponds to the *yz* projection and green corresponds to the *xz* projection.

Following the refinement step and reduction of redundant trajectories, the user selects any number of the returned trajectory sequences to transition into the high-fidelity ephemeris model, as noted in step 8 of Figure 1. A collocation

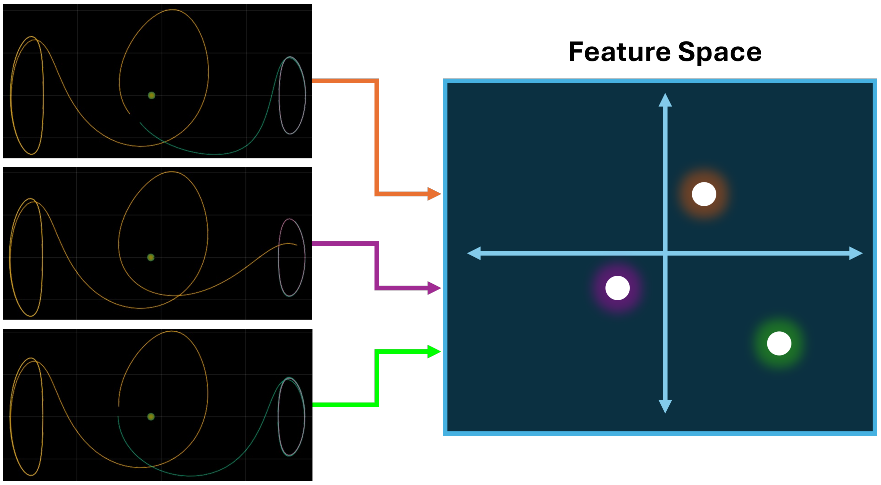


Fig. 7: Representation of trajectories that have similar qualitative properties that are clustered in different groups.

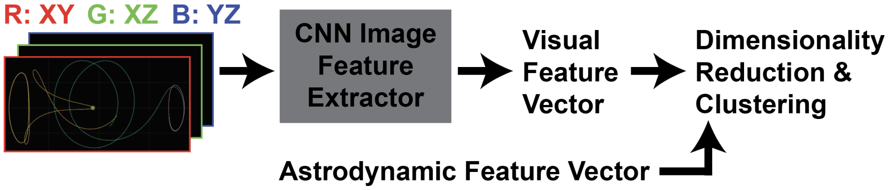


Fig. 8: A high-level illustration of the visual feature vector extraction process.

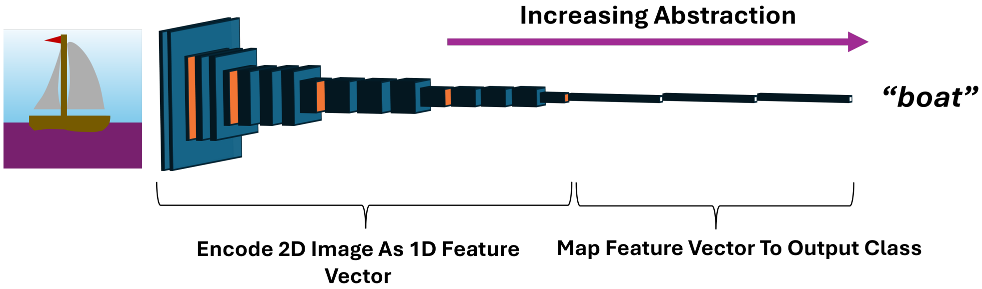
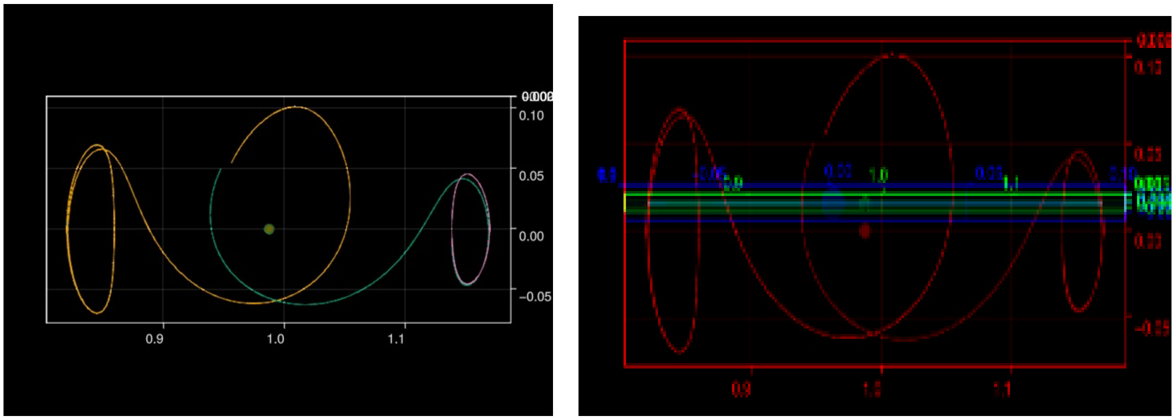


Fig. 9: Illustration of the VGG-16 convolutional neural network, which classifies the image on the left as a boat using various layers of abstraction.

process is used to produce an end-to-end trajectory in the ephemeris model, using maneuvers to connect the segments [18]. The optimization problem is set up to minimize the total mass used when transferring between the departure orbit and the destination orbit, and in this investigation, it was assumed that the spacecraft had a mass of 1000 kg, a specific impulse of 3000 seconds, and a low-thrust engine capable of 1 Newton of thrust. The optimization was performed in FreeFlyer using a 3rd order Legendre-Gauss-Lobatto (LGL) method. The result is the production of multiple, distinct options so decision makers have options to select if other mission constraints – such as fuel usage, time of flight, communication paths, closest approach to the Moon, number of flybys over a lunar region of interest, etc. – also drive the final trajectory chosen for a particular scenario.



(a)

(b)

Fig. 10: (a) Example transfer between L_1 and L_2 Lyapunov orbits produced from graph search. (b) Image generated from example transfer to feed into CNN for further classification.

4. CISLUNAR TRANSFER DESIGN

To demonstrate the path planning algorithm, several example scenarios are considered. First, a classic CR3BP design problem is considered: designing a transfer from an L_1 Lyapunov orbit to an L_2 Lyapunov orbit using their respective stable and unstable manifolds. For orbits at the same Jacobi Constant, it has been demonstrated that heteroclinic transfers, or maneuver free transfers, between planar orbits in the CR3BP are known to exist [3, 7]. This transfer problem has been solved previously with Poincaré maps, where all of the dimensions of a planar problem can be visualized. In another scenario, a transfer can be constructed between orbits at different Jacobi Constants by using a maneuver shift from the unstable manifold of the departing orbit and following the stable manifold of the arrival orbit. The latter scenario is recreated in this analysis; consider an L_1 Lyapunov orbit with a Jacobi Constant equal to 3.1670 as the departure orbit and an L_2 Lyapunov orbit with a Jacobi Constant equal to 3.1666 as the destination orbit. Using the unstable and stable manifolds of these two orbits as the intermediate arcs, several potential transfers between these two orbits are constructed using the path planning algorithm, nine of which are rendered in Figure 11. Note that each of these trajectories have a discontinuity between the blue arc, which is the unstable manifold associated with the L_1 Lyapunov orbit, and the green arc, which is a stable manifold trajectory of the L_2 Lyapunov orbit. Using a collocation process to minimize the propellant usage, a continuous trajectory is produced from the single discontinuous low-fidelity trajectory that was selected by the user for transitioning to a higher-fidelity dynamical model. An example of the initial guess in the CR3BP and the final converged solution in the ephemeris model is rendered in Figure 12. Using the same example, trajectory sequence clustering is applied to the initial guesses, first with astrodynamic feature vectors and then with visual feature vectors, and the two resulting clustering results are presented in Figure 13. With only two returned clusters, neither of which contains a cohesive group of qualitatively equivalent transfers, the astrodynamic feature vector clustering results in Figure 13(a) are clearly not desirable. The visual feature vector clustering results in Figure 13(b) are more desirable, returning instead five clusters with slightly more cohesion among their elements, though the results suggest that further improvements remain necessary to achieve the desired effect of each cluster containing a set of trajectories that appear roughly interchangeable to a human user. To this end, an additional clustering refinement step may be effective to further subdivide each cluster into smaller groups containing more similar trajectories.

Constructing transfers in the planar CR3BP is useful to compare traditional methods, i.e., Poincaré mapping techniques, with the autonomous path planning technique. However, constructing transfer between periodic orbits in the spatial CR3BP is significantly more challenging using traditional methods. The dimensionality of the problem is increased by two (due to the additional position and velocity components in the z-axis direction), making effective visualizations more difficult. The technique presented in this investigation is leveraged to design cislunar transfers in the spatial CR3BP. Consider a scenario where a transfer between a southern L_2 halo orbit and a distant retrograde orbit (DRO) is desired. Designing a transfer between these two orbits has been examined by several investigations

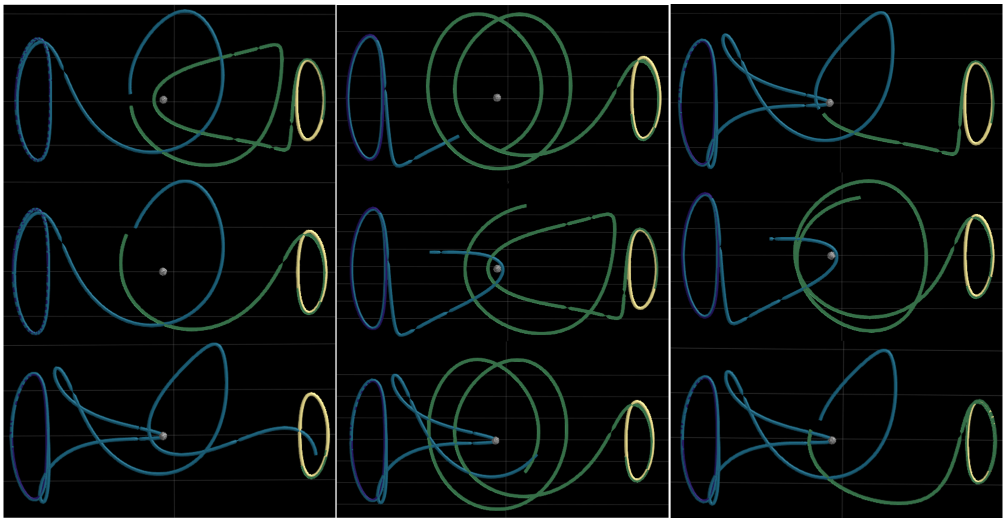
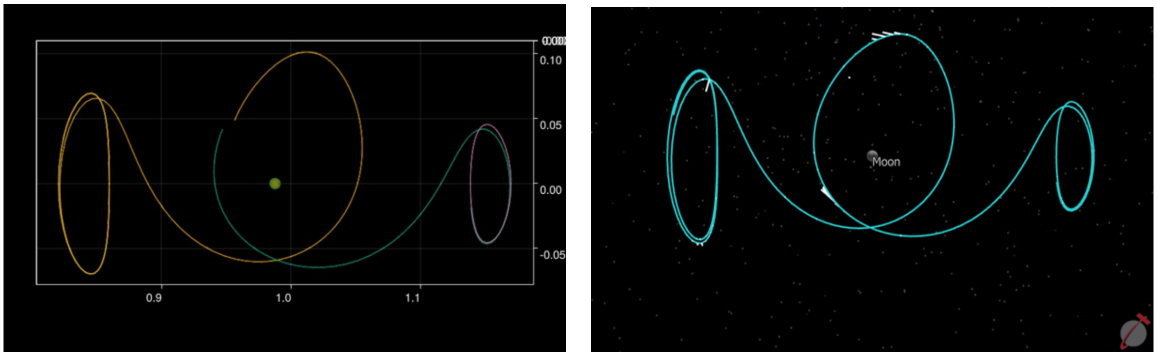


Fig. 11: Various initial guess transfer options between an L_1 Lyapunov orbit and an L_2 Lyapunov orbit.



(a)

(b)

Fig. 12: (a) L_1 Lyapunov orbit to L_2 Lyapunov orbit transfer initial guess using the unstable and stable manifolds as the intermediate arcs. (b) Converged transfer between Lyapunov orbits using a low-thrust engine. The white vectors indicate the direction of thrust. Both trajectories are rendered in the Earth-Moon rotating frame.

and poses challenges given the significant plane change required[8, 9, 20]. Much of the L_2 halo orbit family has a significant out-of-plane component and DROs remain purely in-plane, and traditional plane change maneuvers require a significant amount of Δv to accomplish. To build potential options, several intermediate orbit families and manifolds are used to construct the graph. A subset of the L_2 southern halo family and the respective members unstable orbits, a subset of the L_1 northern halo orbit family unstable manifolds, and members of the 2:1, 3:1, 4:3, and 3:4 planar resonant orbit families were used in the clustering process. Examples of two initial guesses for transfers between the L_2 halo and DRO are rendered in Figure 14 in the Earth-Moon rotating frame. Again, using the initial guess from Figure 14(a), a low-thrust collocation process is leveraged to construct a continuous transfer between the two orbits. The converged orbit is rendered in the Earth-Moon rotating pulsating frame in Figure 15. The general characteristics of the transfer are preserved after convergence using minimal thrust to connect the discontinuous segments.

5. CONCLUDING REMARKS

The algorithm summarized in this investigation seeks to reduce the amount of effort required to construct paths through cislunar space, ideally allowing even a novice trajectory designer to quickly generate high-quality solutions. This investigation augmented previously established processes for constructing cislunar transfer initial guesses through the inclusion of feature vector dimensionality reduction via PCA or UMAP, as well as the introduction a further solution

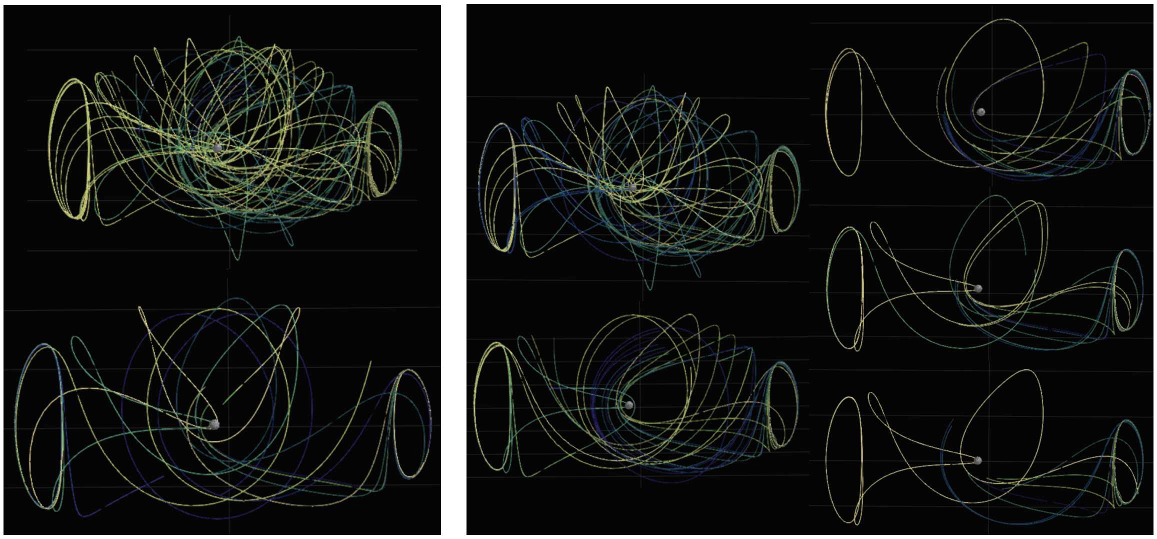
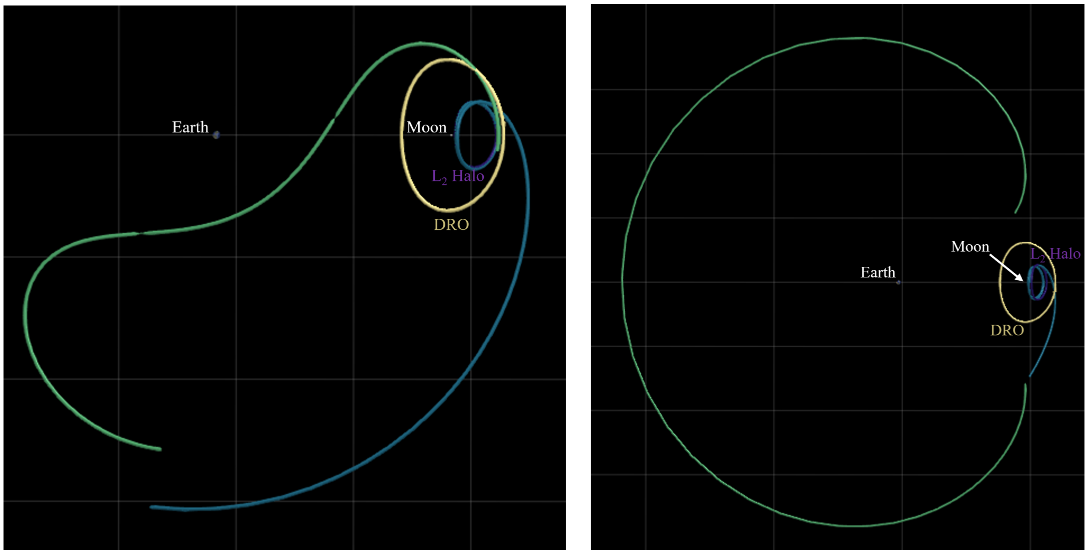

(a) (b)

Fig. 13: (a) Two clusters of trajectory sequences, produced by clustering a set of astrodynamic feature vectors corresponding to initial guess transfer options between an L_1 Lyapunov orbit and an L_2 Lyapunov orbit. (b) Five clusters of trajectory sequences, produced by clustering a set of visual feature vectors corresponding to initial guess options for the same transfer.



(a) (b)

Fig. 14: Two examples of initial guesses for transfers from an L_2 southern halo orbit to a planar distant retrograde orbit (DRO).

clustering step using a visual feature vector from a pre-trained CNN feature extractor. Additionally, this investigation examines how these initial guesses behave when applying low thrust to build an end-to-end trajectory in a Sun-Earth-Moon ephemeris model. Ultimately, a robust method to generate initial guess trajectories streamlines the design process and reduces the expertise required to construct paths in this sensitive regime. Given that cislunar space will be populated rapidly in the coming decades, tools to effectively plan ways to traverse the region will be increasingly necessary. Not only are these methods useful for planning paths for controllable assets, but these same concepts and tools are also important in understanding adversarial assets' movements. Tools that assess where and how quickly

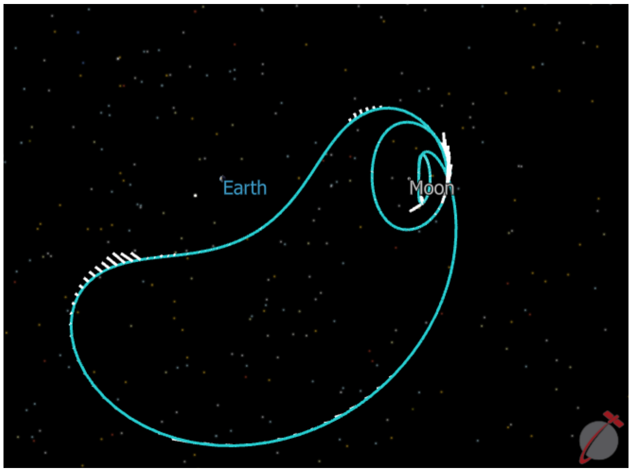


Fig. 15: Converged low-thrust transfer between an L_2 southern halo orbit and a planar distant retrograde orbit (DRO) in the Earth-Moon rotating pulsating frame. The white vectors indicate the direction of thrust.

vehicles could move in cislunar space will be critical to protecting against threats.

6. ACKNOWLEDGEMENTS

The authors would like to thank aiLabs, a.i. solutions' Internal Research & Development Program, for sponsoring this investigation.

REFERENCES

[1] Charu C. Aggarwal. On the Effects of Dimensionality Reduction on High Dimensional Similarity Research. *Proceedings on the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principals of Database Systems*, pages 256–266, May 2001.

[2] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? *Database Theory - ICDT'99*, pages 217–235, January 1999.

[3] R. C. Calleja, E. J. Doedel, A. R. Humphries, A. Lemus-Rodriquez, and B. E. Oldeman. Boundary Value Problem Formulations for Computing Invariant Manifolds and Connecting Orbits in the Circular Restricted Three Body Problem. *Celestial Mechanics and Dynamical Astronomy*, 114(1-2):77–106, October 2012.

[4] Ashwati Das-Stuart, Kathleen C. Howell, and David C. Folta. Rapid trajectory design in complex environments enabled by reinforcement learning and graph search strategies. *Acta Astronautica*, 171:172–195, June 2020.

[5] F. Jorge F. Duarte, Ana L. N. Fred, André Lourenco, and M. Fátima C. Rodrigues. Weighted evidence accumulation clustering. *Fourth Australian Conference on Knowledge Discovery and Data Mining*, 2005.

[6] Carter Franz and Ryan Russell. Database of planar and three-dimensional periodic orbits and families near the moon. *Journal of Astronautical Sciences*, 69:1573–1612, December 2022.

[7] Amanda Haapala and Kathleen C. Howell. A Framework for Construction of Transfers Linking Periodic Libration Point Orbits in the Earth-Moon Spatial Circular Restricted Three-Body Problem. *Journal of Bifurcations and Chaos*, 26(5):1630013–1–1630013–40, May 2016.

[8] Gregory Lantoine. Efficient NRHO to DRO Transfers in Cislunar Space. In *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson, Washington, August 2017.

[9] Melissa McGuire, Laura M Burke, Steve McCarty, Kurt Hack, Ryan Whitley, Diane Davis, and Cesar Ocampo. Low Thrust Cis-Lunar Transfers Using a 40 kW-Class Solar Electric Propulsion Spacecraft. In *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson, Washington, August 2017.

[10] Leland McInnes, John Healy, Saul Nathaniel, and Lukas Grossberger. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[11] Guiliana Miceli, Natasha Bosanac, and Reza Karimi. Generating the Trajectory Design Space for Neptunian System Exploration. In *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, Colorado, August 2024.

[12] Guiliana Miceli, Natasha Bosanac, Jeffrey Stuart, and Farah Alibay. Motion Primitive Approach to Spacecraft Trajectory Design in the Neptune-Triton System. In *AIAA SciTech 2024 Forum*, Orlando, Florida, 2024.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations*, 2015.

[14] Thomas Smith. *Using Motion Primitives to Rapidly Design Trajectories in Multi-Body Systems*. Ph.d. dissertation, University of Colorado Boulder, Boulder, Colorado, 2023.

[15] Thomas Smith and Natasha Bosanac. Motion primitive approach to spacecraft trajectory design in a multi-body system. *Journal of the Astronautical Sciences*, 70(34), September 2023.

[16] M. Song, H. Yang, S.H. Siadat, and M. Pechenizkiy. A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Expert Systems with Applications*, 40:3722–3737, 2013.

[17] Victor Szebehely. *The Theory of Orbits: The Restricted Problem of Three Bodies*. Academic Press, Inc, New York, New York, 1967.

[18] F. Topputo and C. Zhang. Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications. *Abstract and Applied Analysis*, 2014, June 2014.

[19] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37–52, August 1987.

[20] Emily Zimovan-Spreen. *Trajectory Design and Targeting for Applications to the Exploration in Cislunar Space*. Ph.d. dissertation, Purdue University, West Lafayette, Indiana, 2021.