AAS 25-322

SUMMARIZING NATURAL AND CONTROLLED MOTION IN CISLUNAR SPACE WITH BEHAVIORAL MOTION PRIMITIVES

Cole Gillespie[*], Giuliana E. Miceli[†], and Natasha Bosanac[‡]

This paper introduces behavioral motion primitives to summarize the fundamental building blocks of natural and controlled spacecraft trajectories in cislunar space. First, we construct a training data set to capture motions generated for a wide variety of spacecraft parameters and behaviors, i.e., intents and maneuvering objectives. Each trajectory is segmented to produce smaller arcs and then labeled by the governing parameters and behaviors. These arcs are then clustered to automatically discover groups of geometrically similar motions. A representative member is extracted from each cluster and inherits the labels of its members to supply a behavioral motion primitive.

INTRODUCTION

As cislunar space becomes more populated, a wide variety of spacecraft will support science, exploration, infrastructure, and defense operations. These spacecraft will possess a variety of hardware configurations, including their form factor and propulsion system capabilities. Their trajectories will also exhibit a variety of behaviors, composed of their intents, i.e., the itinerary of their trajectory, and maneuvering objective, i.e., their control profile. In the chaotic environment of cislunar space, spacecraft can follow a diverse array of paths to move between two points and these paths depend on spacecraft parameters and control profiles. Accordingly, the solution space is high-dimensional, increasing the complexity of trajectory design, particularly if these parameters may vary. Inverting this problem, paths with a similar geometry may exist across an array of spacecraft parameters and control profiles. This nonuniqueness in solution geometry increases the complexity of using path information to predict possible future motions and estimate the parameters of an unknown spacecraft. In these two scenarios, a digestible summary of this high-dimensional solution space as a function of spacecraft parameters and behaviors may be valuable.

The challenge of summarizing motion in a diverse and high-dimensional solution space that may depend on governing parameters and actions exists across a variety of disciplines. One approach to addressing this challenge leverages the concept of motion primitives, defined in robotics as building blocks of motion.[1] This representation of fundamental motion types can also reframe continuous motion planning as a discrete problem by constructing a motion primitive graph: the nodes are typically individual primitives selected from the library whereas their edges indicate their potential

[*]Graduate Research Assistant, Ann & H.J. Smead Department of Aerospace Engineering Sciences, Colorado Center for Astrodynamics Research, University of Colorado Boulder, Boulder, CO, 80303.
[†]Graduate Research Assistant, Ann & H.J. Smead Department of Aerospace Engineering Sciences, Colorado Center for Astrodynamics Research, University of Colorado Boulder, Boulder, CO, 80303.
[‡]Assistant Professor, Ann & H.J. Smead Department of Aerospace Engineering Sciences, Colorado Center for Astrodynamics Research, University of Colorado Boulder, Boulder, CO, 80303.

for sequential composability. By searching this graph, sequences of motion primitives are then assembled to generate more complex motions.

Across the literature, motion primitives (and similar concepts) have been used for both path-planning and motion prediction. For instance, Frazzoli et al. have used motion primitives to design acrobatic maneuvers for a small helicopter subject to disturbances and sensor noise.[2] Similarly, Majumdar and Tedrake have used motion primitives for online path-planning for an uncrewed aerial vehicle in a dynamic environment with obstacles.[3] Clever et al. have extracted primitives for a robotic humanoid from pregenerated optimally controlled movement sequences and then used these fundamental building blocks to rapidly generate new, complex motions.[4] Inverting this problem, Habibi et al. have also leveraged motion primitives to predict pedestrian motions and update the primitive library as needed.[5] Dermy et al. have also generated robotic movement primitives from a training dataset composed of prior observations and then used these primitives to rapidly predict the robot's intent when completing complex tasks in real-time.[6]

In astrodynamics, motion primitives have recently been used to support spacecraft trajectory design in multi-body gravitational systems. As a proof-of-concept, Smith and Bosanac have used clustering to extract geometrically distinct arcs that supply motion primitives for natural motion along fundamental solutions such as periodic orbits and their hyperbolic invariant manifolds in the circular restricted three-body problem (CR3BP);[7] Miceli et al. have recently extended this approach.[8,9] Smith and Bosanac as well as Miceli et al. have also demonstrated that unique sequences of these motion primitives enable rapid generation of a diverse array of geometrically distinct trajectories that exist across a broader trade space.[8,10] This paper substantially extends prior works in the motion primitive definition and characterization.

This paper introduces behavioral motion primitives to summarize the fundamental building blocks of natural and controlled spacecraft trajectories in cislunar space. Each motion primitive summarizes brief trajectory arcs with similar geometry and is labeled by the associated spacecraft parameters and behaviors that produce those motions. To construct these behavioral motion primitives, we first generate a training dataset that captures a wide variety of motions subject to specific combinations of spacecraft parameters and behaviors. These trajectories are segmented using a geometric sampling scheme to produce smaller arcs and then labeled by their type of motion. These arcs are then clustered to automatically discover groups of geometrically similar motions. A representative member is extracted from each cluster and inherits the labels of all the types of arcs in the cluster to supply a behavioral motion primitive. This primitive may capture an array of possible behaviors and spacecraft parameters producing a similar type of motion. As a result, the behavioral primitives supply two new insights: 1) building blocks of continuous thrust-enabled motion in cislunar space, and 2) an association between distinct types of motion and their current, future, or past behaviors. These insights are valuable for both trajectory design and prediction in cislunar space.

BACKGROUND

Circular Restricted Three-Body Problem

The Earth-Moon Circular Restricted Three-Body Problem (CR3BP) is used to approximate the dynamics governing a spacecraft in cislunar space. This model approximates the Earth (P_1) and Moon (P_2) as traveling on circular orbits around their barycenter with spherically symmetric gravity fields and a constant mass.[11] The mass of the spacecraft is also assumed to be negligible compared to the mass of P_1 and P_2.[11]

When formulating the CR3BP, quantities are typically nondimensionalized using characteristic length, mass, and time quantities. Distances are nondimensionalized using $l^* = 384,400$ km, equal to the average distance between the Earth and the Moon. The characteristic mass $m^* \approx 6.046804 \times 10^{24}$ kg is set equal to the sum of mass of the Earth and the Moon. Finally, the characteristic time $t^* \approx 3.751903 \times 10^5$ s sets the mean motion equal to unity.

Trajectories are often analyzed in the $P_1 - P_2$ rotating frame. This frame uses an origin at the system barycenter. The axes are defined with $\hat{x}$ directed from P_1 to P_2, $\hat{z}$ aligned with the orbital angular momentum of the primary system, and $\hat{y}$ completes the right-handed, orthogonal triad.[11]

Equations of motion for the spacecraft are typically formulated in nondimensional coordinates in the P_1-P_2 rotating frame. The state vector for the spacecraft is first defined as $\bar{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. Then, the equations of motion[11] are written as

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

where U^* is the pseudo-potential function defined as

$$U^* = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2}$$

In these expressions, $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ are the distances of the spacecraft from each of P_1 and P_2 and $\mu \approx 0.012151$ is the mass ratio.

Although there is no analytical solution to the CR3BP, an integral of motion exists. The Jacobi Constant, C_J, is an energy-like constant defined as

$$C_J = 2U^* - \dot{x}^2 - \dot{y}^2 - \dot{z}^2 \tag{3}$$

This quantity can be used to describe spacecraft motion and behaviors.[11]

The CR3BP produces an autonomous dynamical model with fundamental solutions such as libration points, periodic orbits, and hyperbolic invariant manifolds.[11] Libration points are stationary solutions in the rotating frame. Of the five libration points in the system, three are collinear along $y = z = 0$: L_1 is located between the Earth and the Moon, L_2 is located on the far side of the Moon, and L_3 is on the opposite side of the Earth from the Moon. Periodic orbits are defined as trajectories that repeat their motion in their rotating frame after a minimal period P. Some periodic orbits can be generated from the libration points, such as the Lyapunov orbits, whereas others can be identified via bifurcations along other families, such as the halo orbits. Finally, hyperbolic invariant manifolds capture trajectories that asymptotically leave (unstable manifold) or approach (stable manifold) an equilibrium point, periodic orbit, or quasi-periodic orbit.

Thrust-Enabled Motion

To describe the dynamics of a spacecraft with continuous thrust, the equations of motion for the CR3BP are augmented with an additional acceleration. In this paper, the spacecraft is assumed to possess a propulsion system with a constant thrust and constant specific impulse. In addition, the spacecraft mass m is augmented to the state vector. With these definitions, the equations of motion for the thrust-enabled spacecraft are expressed as

$$\ddot{x} = 2\dot{y} + \frac{\partial U^*}{\partial x} + \frac{Tu_x}{m}, \quad \ddot{y} = -2\dot{x} + \frac{\partial U^*}{\partial y} + \frac{Tu_y}{m}, \quad \ddot{z} = \frac{\partial U^*}{\partial z} + \frac{Tu_z}{m} \quad \dot{m} = -\frac{T}{I_{sp} g_0} \tag{4}$$

where $\hat{u} = [u_x, u_y, u_z]$ is the thrust direction, T is the thrust magnitude, I_{sp} is the specific impulse of the thruster, and g_0 is the gravitational acceleration on the surface of the Earth.[12] The fourth component of Equation 4 captures the mass decrement due to the use of a propulsion system.

In this paper, thrust vectors are defined using constant components in the velocity-normal-conormal (VNC) axes relative to the Moon. Although originally derived from the two-body problem, these axes offer meaningful heuristics for maneuver design. These axes are defined with the axis $\hat{V}$ parallel to the instantaneous velocity direction assuming an inertially-fixed observer, $\hat{N}$ is aligned with the orbital angular momentum vector of the spacecraft relative to the Moon, and $\hat{C}$ completes the right-handed triad. These axes become undefined when the position and velocity of the spacecraft relative to the Moon are aligned; this condition may occur as the spacecraft travels sufficiently far from the Moon. Once the thrust unit vector has been specified in the VNC axes relative to the Moon, it is transformed into the rotating frame at each instant of time during numerical integration to enable evaluation of Equation 4.

Differential Geometry

Differential geometry supports describing and analyzing spatial curves using information about the position vector and its time derivatives. At any instant of time along a trajectory, the position, velocity, and acceleration vectors are equal to $\bar{r}(t) = [x(t), y(t), z(t)]^T$, $\bar{v}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$, and $\bar{a}(t) = [\ddot{x}(t), \ddot{y}(t), \ddot{z}(t)]^T$, respectively. Over the time interval $t \in [t_0, t_f]$, measured from a reference initial state, the trajectory traverses a distance equal to the arclength[13]

$$s = \int_{t_0}^{t_f} ds = \int_{t_0}^{t_f} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} dt \tag{5}$$

At any location along the trajectory, the curvature captures the extent to which the path curves away from a straight line. Curvature is calculated as[14]

$$\kappa(\bar{x}) = \frac{\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2 + (\dot{y}\ddot{z} - \ddot{y}\dot{z})^2}}{(\dot{x} + \dot{y} + \dot{z})^{3/2}} \tag{6}$$

Maxima in the curvature indicate the most geometrically meaningful locations along a trajectory.[14] In a multi-body system, these maxima are often located near apses defined relative to useful reference locations such as primary bodies or equilibrium points.[15] The curvature maxima are identified at locations where $\dot{\kappa} = 0$ and $\ddot{\kappa} < 0$.

Density-Based Clustering

Clustering is used to group similar data points together and separate them from dissimilar points.[16] This paper leverages density-based clustering to discover sufficiently dense groupings of data in a finite-dimensional feature vector space. This paper uses two density-based clustering algorithms: 1) Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[17] and 2) Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN).[18]

DBSCAN was developed by Ester, Kriegel, Sander, and Xu to group data with overlapping neighborhoods of a specified size.[17] A core point is defined as a member with an m_{pts}-neighborhood with a radius below a specified size ϵ. A border point lies within the m_{pts}-neighborhood of a core point, but its own m_{pts}-neighborhood size is larger than ϵ. A noise point, however, does not satisfy this criterion. Then, each cluster is composed of core points that have overlapping neighborhoods as

well as their border points. DBSCAN is useful when suitable values of m_{pts} and ϵ are known for a given dataset. In this work, DBSCAN is accessed through MATLAB.

HDBSCAN, developed by Campello, Moulavi, and Sander,[18] first transforms the data to capture proximity between two members of the dataset as well as the density of their local neighborhoods. The mutual reachability distance is defined as the maximum of the distance between the two points, the distance from the first point to its m_{pts}-neighbor, and the distance of the second point to its m_{pts}-neighbor. Using this new representation of distance between all members of the dataset, a minimum spanning tree is generated.

HDBSCAN then selects clusters using a hierarchy of all possible results. First, the minimum spanning tree is converted into a cluster hierarchy that is then condensed into a smaller tree. Each node in this condensed tree represents a cluster with at least the minimum number of points, $m_{min,clust}$. As the density level increases over the tree, each cluster either shrinks, disappears, or splits into two if each sub-cluster has more than the minimum number of points. With this condensed tree, clusters are selected to maximize the stability based on an excess of mass definition.[18] The output of HDBSCAN is an assignment for every data point to either a cluster or noise. Noise points are those that do not meet the density threshold to fit into any cluster. Due to its hierarchical approach, HDBSCAN is useful when the expected clusters possess distinct densities and sizes as well when a suitable neighborhood size is not known a priori.

The clusters generated by HDBSCAN depend on a few key governing parameters. The minimum number of members needed to form a cluster is governed by the parameter $m_{min,clust}$ whereas the number of neighbors used to define density is selected using m_{pts}.[18] Smaller values of these two parameters tend to produce more local clusters. HDBSCAN has also been modified to use a distance threshold epsilon, ϵ, below which two neighboring data points are grouped together.[19] In this work, HDBSCAN is accessed using the *hdbscan* Python library.[20]

TECHNICAL APPROACH

Defining a Behavioral Motion Primitive

Spacecraft operating in cislunar space can follow a wide variety of paths throughout the system, influenced by their properties and behavior. This paper introduces the concept of a behavioral motion primitive to summarize a diverse set of spacecraft motions in a continuous system using a discrete representation of fundamental building blocks. Accordingly, this definition builds upon prior work by Smith and Bosanac[10] as well as Miceli and Bosanac[8] in applying the concept of a motion primitive, most commonly used in robotics, to spacecraft trajectories. However, the key element of this new definition is encoding information about the ranges of parameters or behaviors used to generate each motion type. As a result, a behavioral motion primitive conveys both a geometrically distinct type of motion and an array of associated spacecraft parameters or behaviors.

In this paper, the behavior of a spacecraft includes its intent (i.e., its goal) and the maneuvering objective (i.e., approach used to achieve that goal). In this proof of concept, intents include following bounded motion, approaching or departing a bounded motion, or transferring between two bounded motions. Maneuvering objectives considered in this paper include not maneuvering, i.e., natural motion, and applying continuous thrust using a prespecified control law. Note that one motion type may be possible with a wide variety of behaviors. For instance, an array of different thrust vectors may produce geometrically similar motions, although the exact path differs for the same spacecraft and propulsion system.

Step 1: Generating Training Data

Training data is generated to represent an array of expected motions for a spacecraft exhibiting various behaviors in the lunar vicinity. In this paper, these data include both natural trajectories and continuous thrust-enabled motions near the Moon. For continuous-thrust trajectories, one control law is specified a priori and thrust directions are discretely sampled. This approach produces a large and diverse array of possible trajectories.

Natural, Bounded Motions In this proof of concept, periodic orbits are used to generate spacecraft motions that are naturally bounded. The periodic orbits used in this paper include well-known planar and spatial libration point orbits near L_1 and L_2 (Lyapunov, halo, axial, and vertical orbits) as well as Moon-centered orbits (distant prograde orbits, distant retrograde orbits, and low prograde orbits). Depending on the family, approximately 250 to 2000 members are sampled. Although there are more complex periodic orbits and quasi-periodic orbits that also supply natural, bounded motions, this subset offers a sufficiently diverse and useful starting point for demonstration.

Naturally Approaching and Departing Bounded Motions To capture trajectories that naturally approach or depart a periodic orbit in the CR3BP, the unstable and stable manifolds of unstable periodic orbits are generated. Along each periodic orbit family, approximately 20-80 members are sampled at Jacobi constants between 3.0 and 3.2 to bound the analysis. Each unstable periodic orbit is sampled using N states equally spaced along its arclength. After stepping into the stable or unstable eigenspace, the associated trajectories are generated until they either impact a primary, exceed a non-dimensional distance from the Moon equal to unity, or exceed a maximum propagation time. The maximum propagation time t_{prop} is set equal to 3 months after the perturbation-doubling time, $t_{pd} = \ln(2)/\ln(\lambda_{max})P$, where λ_{max} is the eigenvalue of the monodromy matrix associated with the unstable mode and the largest magnitude.[21] Thus, $t_{prop} = t_{pd} + 3$ months. When two stable or unstable modes exist, each mode is excited independently to produce $2N$ trajectories in either the stable or unstable manifold.

Naturally Transferring Between Bounded Motions Natural transfers are generated as paths where a spacecraft asymptotically approaches periodic orbits in forward and backward time. These trajectories are labeled heteroclinic and homoclinic connections if they approach two distinct orbits or the same orbit, respectively.[22] Furthermore, these transfers can predict nearby finite-time transfers with low maneuver requirements. To generate these transfers, initial guesses are found using a Poincaré map with a surface of section at either the x-coordinate or y-coordinate of the Moon, depending on the desired transfer geometry. Stable and unstable manifolds of the selected initial and final orbits are propagated backward and forward in time, respectively. Close crossings of these manifolds with the surface of section are then used to manually identify initial guesses for heteroclinic or homoclinic connections. By using different combinations of crossings, geometrically distinct initial guesses are identified. These initial guesses are then corrected via multiple-shooting, and then natural parameter continuation is used to produce a family of solutions.

Continuous-Thrust Approaches and Departures to/from Bounded Motions In this proof of concept, the spacecraft is assumed to be an ESPA-compatible SmallSat with a mass of around 180 kg.[23,24] However, the mass is allowed to be slightly higher when generating trajectories that approach a periodic orbit. This SmallSat is equipped with a propulsion system that can impart a thrust magnitude of 13 mN with an I_{sp} of 1390 s.[24,25] Furthermore, the thrust direction is specified as a constant unit vector in the VNC axes defined relative to the Moon. Although there are a wide variety of possible control laws, this one description offers a useful starting point for demonstration and a

connection to well-known maneuvering heuristics when the spacecraft remains sufficiently close to the Moon. The thrust direction is parameterized using two angles as

$$\hat{u} = \cos(\theta_{VC})\cos(\theta_N)\hat{V} + \sin(\theta_{VC})\cos(\theta_N)\hat{C} + \sin(\theta_N)\hat{N} \tag{7}$$

These two angles are sampled in 90-degree increments to generate distinct thrust vector directions and, therefore, maneuvering objectives.

Continuous-thrust trajectories are generated by sampling states along periodic orbits and propagating forward or backward in time with a specified control law and spacecraft. First, each periodic orbit is sampled using N states that are equally spaced along its arclength. The thruster is then activated and continuous-thrust trajectories are generated by propagating forward and backward to produce paths that depart or approach the periodic orbit with a mass of 180 kg. Of course, the mass of the spacecraft changes along these trajectories, meaning that the spacecraft may begin with a higher or lower mass before reaching the periodic orbit. Similar to the natural asymptotic approaches and departures from periodic orbits, these trajectories are generated until they either impact a primary, exceed one non-dimensional length unit away from the Moon, or exceed a maximum propagation time of three months.

Step 2: Discretizing and Describing Arcs within Training Dataset

To support discovering building blocks of motion, each approach, departure, or transfer trajectory from the training dataset is discretized into smaller arcs before clustering and extracting the primitives. The purpose of discretization is to ensure that each arc is appropriately compared with other arcs, regardless of prior or future behavior. For instance, if two heteroclinic connections are similar but one performs a revolution around the Moon while the other does not, discretization allows the similar approaching/departing arcs of both transfers to be grouped together while differentiating the remaining part. However, periodic orbits are not separated into arcs and, therefore, their associated primitives capture motion for a full orbit period.

Each approach, departure, or transfer trajectory is discretized into smaller arcs in a manner that captures their geometry. Specifically, each arc is defined to begin at every maximum in the curvature along the selected trajectory. Each subsequent arc is then offset from this previous arc by a single maxima. Then, the arc is defined to encompass a total of five maxima in curvature, including the initial point. Due to the correlation between maxima in the curvature and apses, this number of curvature maxima often corresponds to trajectories that complete up to two revolutions within the evolving osculating plane. Of course, some trajectories are geometrically more complex and may encompass fewer revolutions or trace out smaller arcs. Examples of the types of arcs that may be constructed using this definition are displayed in Figure 1 with segments that include a) two curvature maxima (initial state included), b) five curvature maxima, and c) eight curvature maxima; the red markers locate the curvature maxima. Using arcs that encompass five curvature maxima produces "building block" sized segments of motion and supports geometric differentiation. If the arc terminates after four maxima due to impacting a primary or exceeding the distance threshold, its final state is used in place of the fifth curvature maximum. If, however, a trajectory does not produce any arcs with at least 4 maxima in the curvature, it does not supply a usable arc. Furthermore, in cases where segments exceed an arclength threshold, they are also discretized by position extrema. This logic is useful in spiral in/out cases where there are no curvature maxima over extended time intervals.

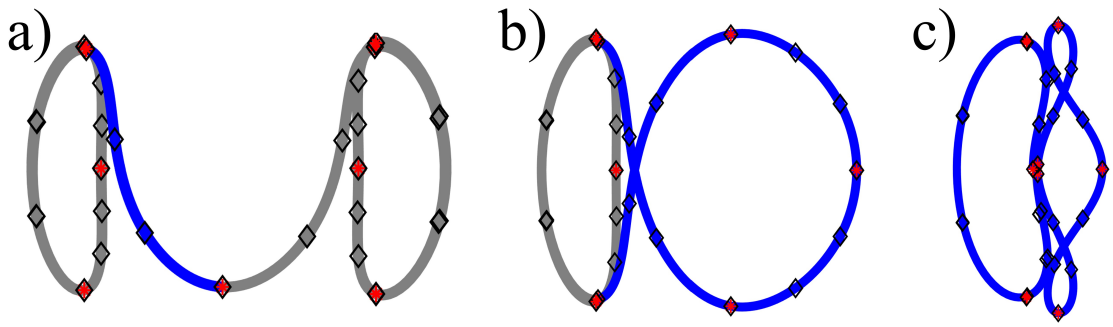

Figure 1. Examples of arcs defined to encompass a distinct number of maxima in the curvature (red diamonds, initial state included): a) 2, b) 5, c) 8. Black diamonds located samples evenly distributed in arclength between subsequent maxima.

Each arc is then sampled to support summarizing a continuous trajectory. Four segments are defined between each subsequent pair of curvature maxima. Then, samples are placed at each of the extrema as well as two states evenly distributed in the arclength between them, represented by the black diamonds in Figure 1. This sampling approach adapts the spacing between states to the complexity and sensitivity of each segment, without being biased by differences in speed along a trajectory. Furthermore, by spacing the arclength samples between the extrema samples, similar arcs have samples in similar places. As a result, each arc is described by 13 samples.

Each continuous arc along an approach, departure, or transfer trajectory is described by finite-dimensional feature vectors that are used to support clustering. These feature vectors are defined using state information, i.e., position and velocity, at the samples. A position-based feature vector is defined as

$$\bar{f}_p = [x_1, y_1, z_1, ..., x_{13}, y_{13}, z_{13}] \tag{8}$$

In addition, a shape-based feature vector is defined as

$$\bar{f}_v = [\tilde{\tilde{x}}_1, \tilde{\tilde{y}}_1, \tilde{\tilde{z}}_1, ..., \tilde{\tilde{x}}_{13}, \tilde{\tilde{y}}_{13}, \tilde{\tilde{z}}_{13}] \tag{9}$$

where $\tilde{\tilde{x}}$ are the components of the velocity unit vector at each state. Using the velocity unit vectors ensures that each component is on a similar scale, regardless of the sensitivity of the region where the spacecraft is located. Of course, this definition does suffer from a singularity if the speed is equal to zero; however, this case is not encountered in the training dataset.

Step 3: Extracting Motion Primitives from Training Data

The arcs generated for each behavior in the training dataset are clustered to produce groups of geometrically similar paths. For periodic orbits, an analytical criterion is used to separate the members along each family into continuous groups. However, for arcs that are generated from approach, departure, or transfer trajectories, clustering algorithms are used to perform an unsupervised grouping using their feature vectors. To produce high-quality clusters from a diverse array of trajectories, this clustering is performed in two steps using a process that was developed by Bosanac.[26]

Grouping Periodic Orbits In the case of bounded motion, groups of periodic orbits along a continuous family are constructed using an analytical approach. Recall that the instantaneous value of the curvature captures the shape of a nonlinear path at the associated state. As the geometry of a

trajectory evolves, the time history of the curvature also evolves. These changes in the geometry can, therefore, be identified using stationary points in the curvature. In this paper, the number of maxima in the curvature is calculated along each discretely sampled periodic orbit along a family. When the number of maxima in the curvature changes between two neighboring, discretely sampled members, a change in the geometry also occurs. Thus, each group of geometrically similar periodic orbits is constructed as neighboring, discretely sampled members between subsequent changes in the number of maxima in the curvature. This approach produces groups of periodic orbits that correspond to individual segments of a family with a similar geometry.

Initial, Coarse Clustering A diverse set of approach, departure, or transfer arcs associated with a single behavior is first coarsely clustered by their shape using $\bar{f}_v$ and the HDBSCAN clustering algorithm. This algorithm enables the discovery of groupings of data with uneven shapes, densities, and membership sizes.[18] During this clustering step, the hyperparameters are selected to prioritize producing smaller local clusters along with a Euclidean distance measure. In this research, values of $m_{min,clust} = 5$ and $min_{pts} = 4$ were used. To specify a threshold on the neighborhood size used to distinguish two different clusters, the value of ϵ is defined heuristically as $\epsilon = 2\sqrt{13}\ \sin(\Delta\theta/2)$ where $\Delta\theta = 10^\circ$ is a threshold on the average angle between two unit vectors, and scaled by the 13 nodes in the trajectory.[27] Any arcs that are designated as noise are discarded. The output of this step is a rough grouping of trajectories with a similar shape.

Cluster Refinement For more complex sets of arcs that approach or depart bounded motion, a cluster refinement process is valuable to ensure high quality clusters with few, if any, outliers. This approach is modeled after a convoy detection scheme that has been used in the trajectory clustering literature.[28] A convoy detection approach focuses on discovering groups of trajectories that remain density-connected for their entire duration. Inspired by this approach, Bosanac has adapted this idea for refining clusters of spacecraft trajectories.[26]

Cluster refinement based on convoy detection is applied individually to each coarsely grouped set of arcs. First, the ith state along each arc is described by a three-dimensional feature vector; either $\bar{f}_v$ or $\bar{f}_p$. Then, those feature vectors are clustered using DBSCAN with $min_{pts} = 4$, a Euclidean distance measure, and $\epsilon = m_{pts}\max(e, \epsilon_{threshold})$ where e is the median distance between each member and its nearest neighbor and $\epsilon_{threshold}$ is a minimum threshold.[26] This heuristically selected neighborhood size is large enough to enable DBSCAN to detect groups of core points without being biased by outliers. This process is repeated for all 13 states along the arcs. Any arcs that are consistently grouped together in all 13 clustering results and possess at least $m_{min,clust}$ members form a cluster. However, any arcs that are considered noise at any stage are discarded. This process is performed twice: first in the velocity and then in the position-based feature vector space. Applying refinement independently in each feature vector space ensures that each group of arcs possesses a similar shape and path through the configuration space for their entire duration.

Extracting the Motion Primitive A motion primitive is extracted as a representative member of each cluster. Specifically, the representative possesses the minimum value of the maximum distance to any other member of the cluster in a specified feature vector space. This definition is mathematically expressed as

$$\bar{f}_{minOfMax} = \min_{\bar{f}_i \in \mathcal{F}} (\max_{\bar{f}_j \in \mathcal{F}} (||\bar{f}_i - \bar{f}_j||)) \tag{10}$$

where $\mathcal{F}$ is the set of all feature vectors and $\bar{f}_i$ and $\bar{f}_j$ are any two feature vectors in $\mathcal{F}$. The benefit of this approach is that the selected primitive is not as easily skewed by regions of higher density within the cluster and instead selects a centrally located point. This definition is compared to the

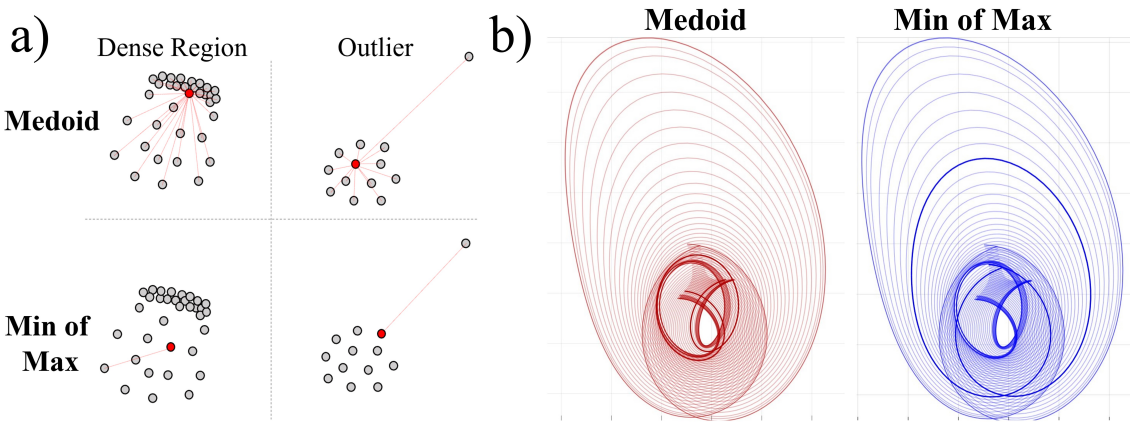

Figure 2. Two approaches to calculating a motion primitive: a) conceptual representation, and b) example for planar cluster.

well-known medoid, for example, in Figure 2 a). Note that in part b), there is a dense region towards the center.

Visualizing Clusters Clusters are typically visualized by displaying a subset of members of the cluster along with the primitive. The primitive is usually bolded or darker in color for visual clarity. In cases where the cluster possesses more than 20 members, only a subset of up to 20 members that are approximately evenly distributed in the shape-based feature vector space are displayed. This configuration supports visualizing the region of the phase space encompassed by the cluster in a digestible manner.

Step 4: Labeling Motion Primitives by Behavior

Once all the clusters have been generated to summarize arcs with a specific behavior, they are labeled by these behaviors. These labels are critical for design or prediction due to the extra context and filtering they provide and are a key component of constructing behavioral motion primitives. The label for each arc is made up of several components. One example is the intent of the spacecraft, which includes the following categories: approach/depart, escape, bounded motion, or reconfiguration. If the motion approaches and/or departs from an orbit, the label also describes the initial and/or final orbits. The motion type field captures if the motion is natural or controlled. When a continuous thrust is applied, values for the thrust magnitude, I_{sp}, thrust direction and definition, and spacecraft mass at the initial/final orbit are included. As an example, a spacecraft traveling along a periodic orbit would be labeled with the intent of natural, bounded motion along with the specific family and member in that family. An overall representation of the label structure and its entries is provided in Table 1.

Step 5: Aggregating Behavioral Motion Primitives Across Training Dataset

Once clusters and motion primitives have been generated for each individual behavior, i.e., approaching or departing a specific motion or thrusting in a specific direction, they are then aggregated across distinct behaviors to summarize common motion types. For instance, arcs along an unstable manifold of one periodic orbit may be geometrically similar to arcs along a stable manifold of another periodic orbit. Thus, the two groups can be merged into one single cluster. In that case, one

Table 1. Labeling convention for behavioral motion primitives

Family	L_1 Lyapunov, DPO, L_2 axial, etc.
Reference periodic orbit state	$[x, y, z, \dot{x}, \dot{y}, \dot{z}]$
Reference periodic orbit period	(double)
Reference periodic orbit energy	(double)
Motion type	natural or controlled
Thrust magnitude (N)	(double)
Specific impulse (s)	(double)
Thrust vector direction	$\hat{u}$
Thrust vector definition	basis vectors
Boundary condition mass (kg)	(double)
Intent	approach, depart, bounded, or escape
Departure orbit	N/A, L_1 Lyapunov, DPO, etc.
Approach orbit	N/A, L_1 Lyapunov, DPO, etc.

primitive may describe a type of motion associated with multiple distinct behaviors.

Aggregating clusters across two distinct behaviors consists of two key steps. First, for each cluster summarizing motion in one dataset or behavior, its k-nearest neighboring clusters from the other dataset or behavior are coarsely identified. To implement this step in a computationally efficient manner, these neighbors are identified using the position-based feature vectors $\bar{f}_p$ of the motion primitives of each cluster. These neighboring clusters supply candidates for aggregation. To determine if each candidate pair of clusters and primitives should be merged, the cluster refinement process is applied to members of both groups. If any members from each group are consistently clustered together in each of the shape and position-based feature vector spaces, these clusters are aggregated to form a new, merged cluster. An example of this process is conceptually depicted in Figure 3. Members of these new aggregated global clusters then inherit the labels from their original clusters and a new behavioral motion primitive is calculated. This aggregation process is performed hierarchically: 1) first across distinct behaviors for a single reference family, e.g., aggregating trajectories that approach an L_1 Lyapunov orbit either naturally or with an array of thrust directions, and then 2) across families. The result of this aggregation step is a single global cluster summary.

RESULTS

The presented technical approach is applied to a diverse array of trajectories across a variety of behaviors. The result is a large and diverse training dataset and, therefore, a wide array of behavioral motion primitives that summarize building blocks of motion for natural and controlled trajectories beginning in the vicinity of the Moon in the Earth-Moon CR3BP. The training data and the number of clusters generated for each intent associated with a specific family are summarized in Table 2 and Table 3. For the controlled motion, data is generated for all 6 thrust-vector directions for the spacecraft described earlier. This section presents selected examples for each intent.

Bounded Motion

Periodic orbits in the CR3BP are summarized to supply behavioral motion primitives for natural, bounded motion. Figure 4 displays selected examples of clusters of trajectories that resemble a motion primitive and its labels. In this figure, each image includes the primitive plotted with thick

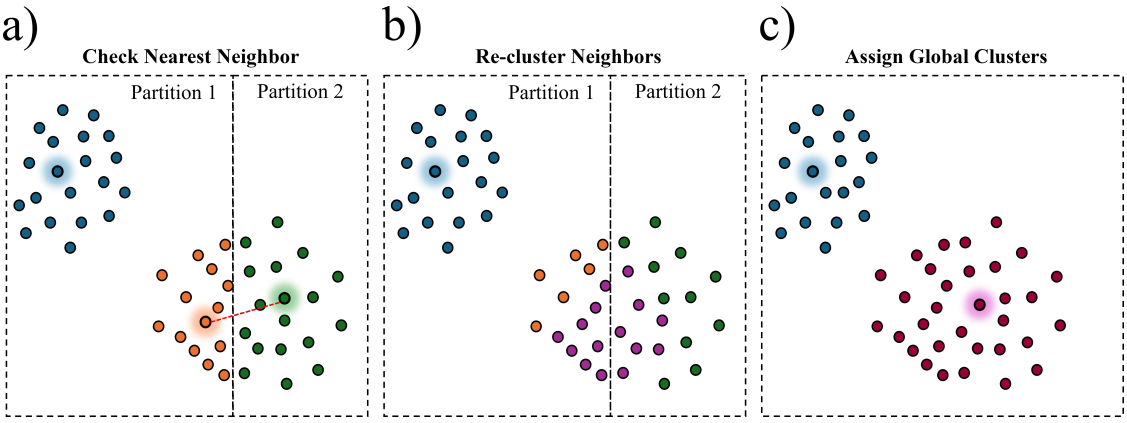

Figure 3. Conceptual depiction of cluster aggregation: a) identifying candidates for merging, b) clustering candidate clusters, and c) merging to form a global cluster.

Table 2. Type of data in training set and number of behavioral motion primitives prior to aggregation across the entire dataset

	Number of Primitives		
Family	**Bounded**	**Approach or Depart**	
		Natural	**Total (natural and/or thrust-enabled)**
L_1 Lyapunov	4	768	1504
L_1 halo	5	1283	4020
L_1 axial	2	1020	2175
L_1 vertical	2	1083	2823
L_2 Lyapunov	3	830	1752
L_2 halo	5	588	2089
L_2 axial	2	570	1446
L_2 vertical	2	591	1820
P_2 DPO	7	782	1734
P_2 DRO	2	N/A	N/A
P_2 LoPO	4	N/A	N/A

blue curves and subsampled members of the clusters plotted with thinner blue curves. Red dots along the primitive show the maxima in curvature. Each gray box lists the labels for each behavioral motion primitive as well as the ranges of values of the orbit period and Jacobi constant spanned by members of the cluster. For primitives summarizing members of the same family, each motion type corresponds to trajectories exhibiting a distinct number of curvature maxima and, therefore, a distinct geometry.

Naturally Approaching or Departing Bounded Motions

A diverse array of trajectories that naturally approach or depart periodic orbits are also summarized. These trajectories are first generated and clustered in smaller subsets that correspond to departing or approaching a single periodic orbit. Then, the clusters are aggregated across the entire

Table 3. Number of behavioral motion primitives in reconfiguration dataset

Departing Family	Approaching Family	Number of Primitives
L_1 Lyapunov	L_1 Lyapunov	330
L_1 Lyapunov	L_2 Lyapunov	264
L_1 Lyapunov	P_2 DPO	126
L_2 Lyapunov	L_1 Lyapunov	136
L_2 Lyapunov	L_2 Lyapunov	361
L_2 Lyapunov	P_2 DPO	24
P_2 DPO	L_1 Lyapunov	239
P_2 DPO	L_2 Lyapunov	30
P_2 DPO	P_2 DPO	178

set of natural motions that approach and/or depart any periodic orbit.

Selected examples of the behavioral motion primitives with this type of intent are displayed in Figure 5, along the period of the reference orbit, labeled as 'Ref P' and expressed in nondimensional units. In this figure, any additional black curves are a projection of the motion primitive onto a plane parallel to the x-y plane, but displaced in the z coordinate. This curve supports a better interpretation of trajectories with a substantial spatial component. In each subfigure, the large blue marker displays the initial condition to indicate direction of motion. The selected examples include behavioral motion primitives that summarize arcs approaching or departing a range of periodic orbits. Finally, from the clusters in Figure 5, it is clear that even as motion becomes more complex, the clusters contain geometrically similar motions.

Thrust-Enabled Motion

When thrust-enabled trajectories are included in the data, it can become more challenging to visualize the clusters and their associated behaviors. Accordingly, each figure also summarizes the array of thrust vectors that lead to geometrically similar motions. As depicted in Figure 6, a sphere depicts the possible thrust directions in the VNC axes, with each line indicating a thrust vector in that direction produces members included in the selected cluster. To facilitate clear visualization, these arrows are also colored: $\hat{u} = \pm\hat{V}$ is colored red, $\hat{u} = \pm\hat{N}$ is colored blue, and $\hat{u} = \pm\hat{C}$ is colored magenta. A black circle in the center as well as a thin black border around the thrust direction figure indicates that natural motion is also present in the cluster.

Several examples of behavioral motion primitives of thrust-enabled trajectories and their geometrically similar motions are displayed in Figure 6. Each behavioral motion primitive may capture trajectories that begin with a distinct mass or use a different thrust vector. Yet, their motions are geometrically similar. This figure also demonstrates that a wide array of geometries are possible for spatial, thrust-enabled motion in the CR3BP.

Naturally Reconfiguring Between Bounded Motions

Natural reconfiguration is approximated using heteroclinic and homoclinic transfers. However, the behavioral motion primitives reveal that it may not be necessary to compute and summarize these motions directly, saving substantial analyst time and workload when forming a training dataset. In the case of a heteroclinic connection, these trajectories lie in both the unstable manifold of one

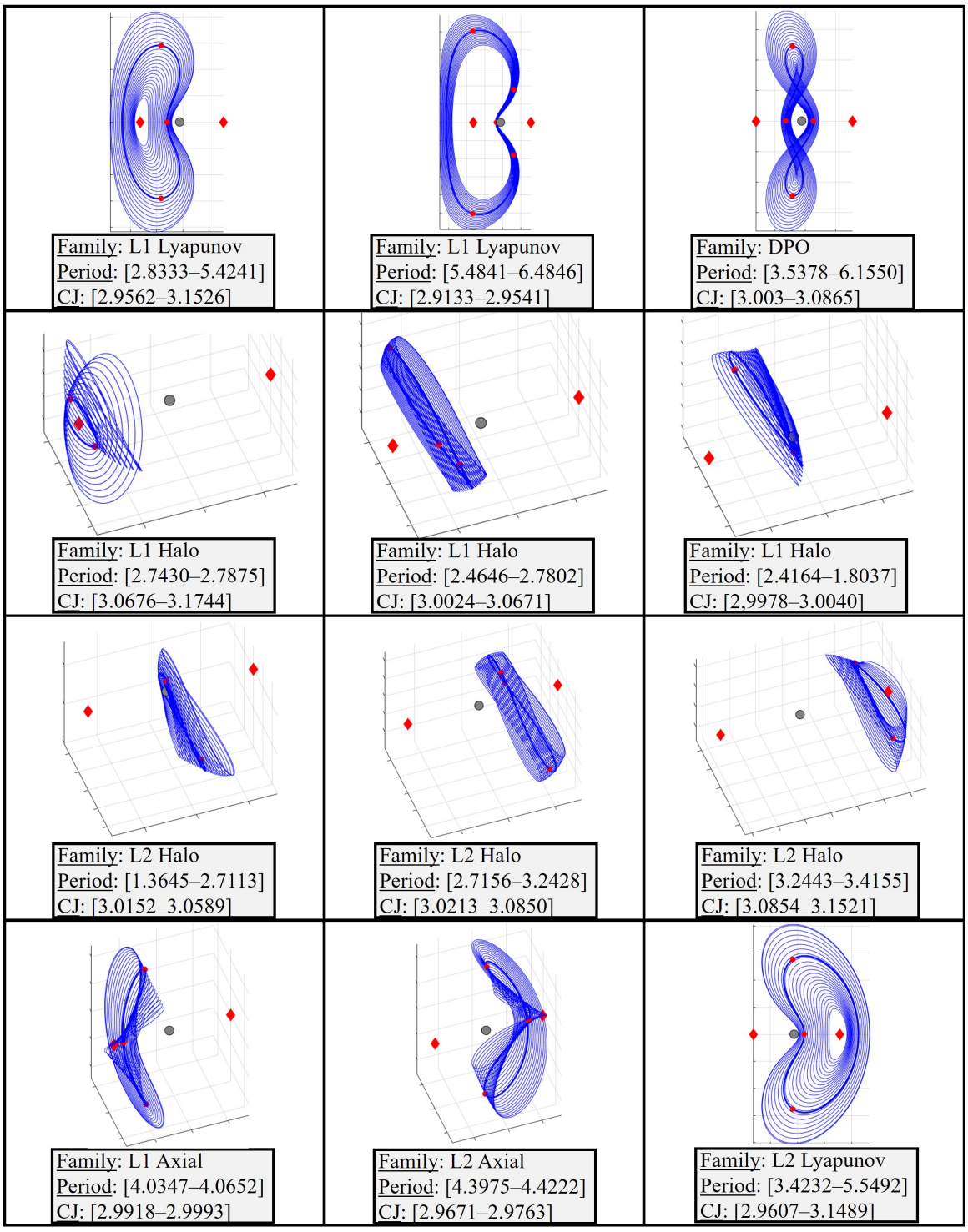

Figure 4. Behavioral motion primitives for naturally bounded motion in the CR3BP.

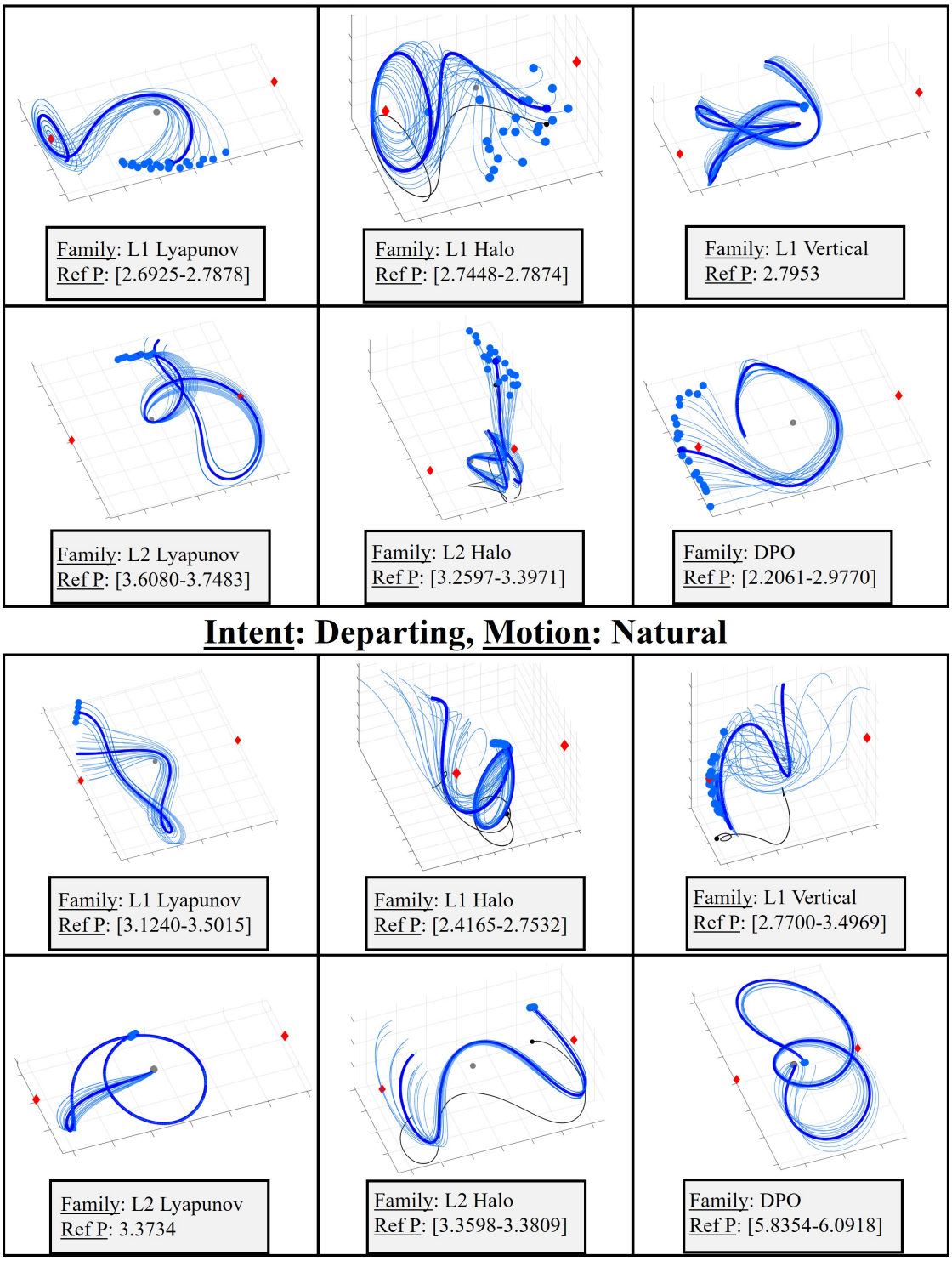

Figure 5. Behavioral motion primitives summarizing motion that naturally approaches or departs from bounded motions in the CR3BP.

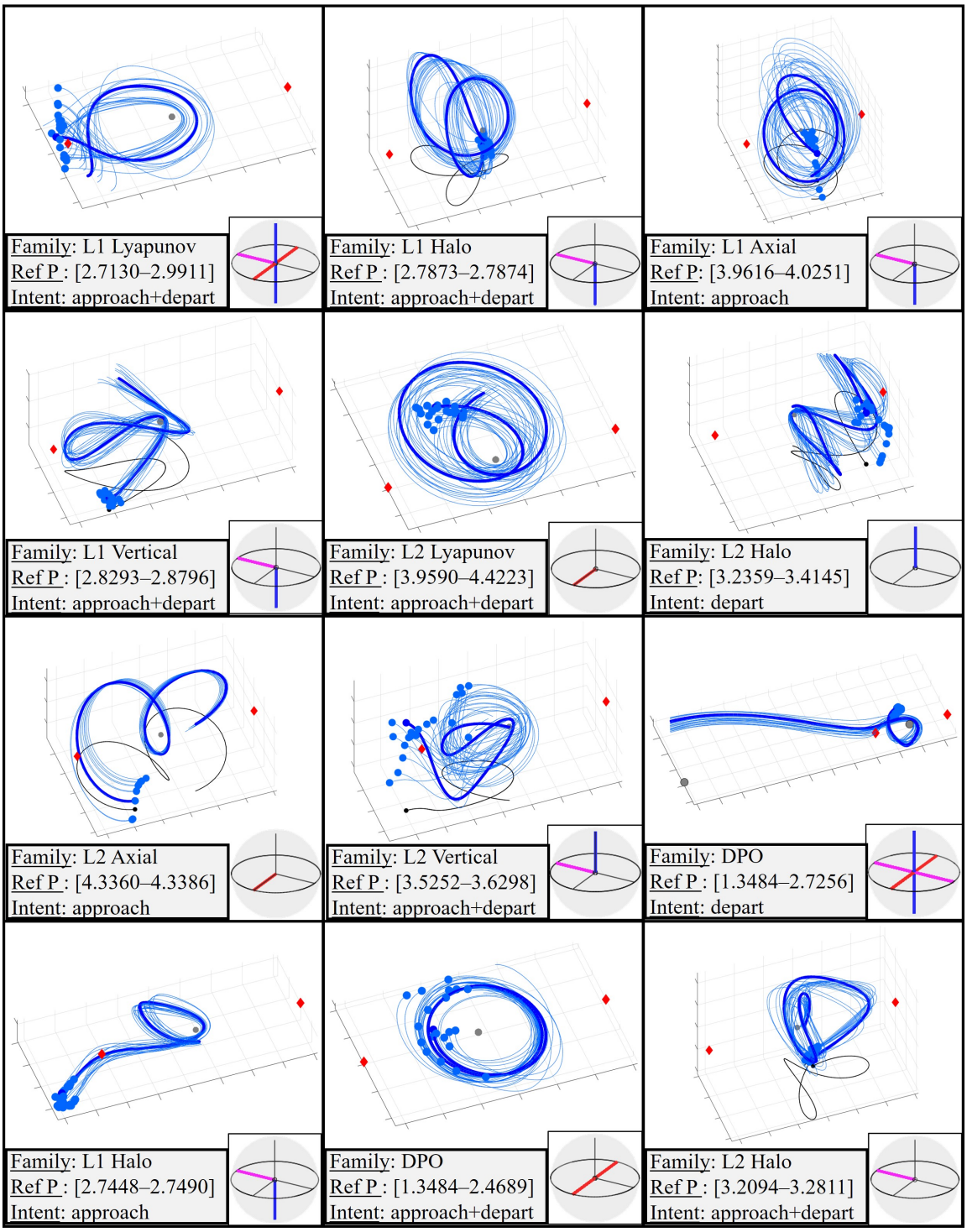

Figure 6. Behavioral motion primitives summarizing natural thrust-enabled approaches into or departures from members of one periodic orbit family in the CR3BP.

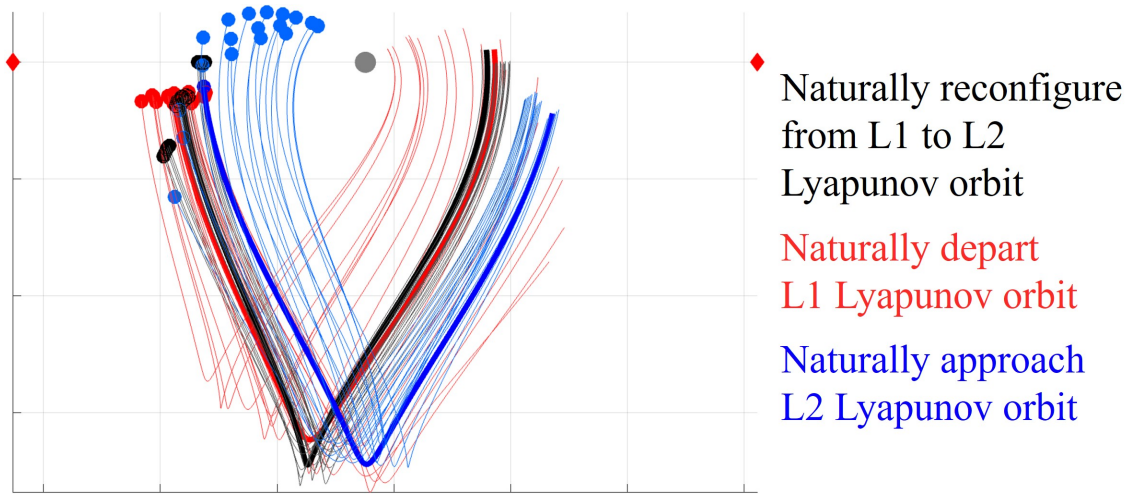

Figure 7. Aggregated behavioral motion primitives resembling arcs along a heteroclinic connection from an L_1 Lyapunov orbit to an L_2 Lyapunov orbit.

periodic orbit and the stable manifold of another orbit. Accordingly, one or more behavioral motion primitives from manifolds of those orbits at nearby energy levels should resemble behavioral motion primitives from the heteroclinic connection. As an example, Figure 7 displays arcs associated with the same type of motion with three distinct behaviors: departing an L_1 Lyapunov orbit in blue with $C_J = 3.1000$, approaching an L_2 Lyapunov orbit in red with $C_J = 3.0946$, and lying on the exact heteroclinic connection in black with $C_J = 3.0957$. The aggregation process groups all of these arcs together to form one global cluster.

Summary of Behavioral Motion Primitive Library

After clusters of geometrically similar trajectories have been aggregated across the entire array of distinct behaviors, an extensive library of behavioral motion primitives is formed. A total of 15,337 behavioral motion primitives are generated. Within this set, 38 capture naturally bounded motion, 14,287 correspond to primitives that approach or depart periodic orbits in a single family, and 1,012 behavioral motion primitives summarize geometrically similar motion that could potentially approach or depart multiple orbit families. A selection of these primitives and geometrically similar trajectories are shown in Figure 8. In this figure, the annotations indicate 1) the family of periodic orbits that the trajectories approach or depart and 2) a summary of whether motion is natural or the associated thrust vectors.

CONCLUSIONS

This paper introduces the concept of a behavioral motion primitive to summarize natural and controlled spacecraft motion in a low-fidelity model of cislunar space. A behavioral motion primitive is defined to capture a single type of motion along with the associated behaviors, including intent and maneuvering objective, and spacecraft parameters. First, training data is generated across a variety of behaviors. Approach, departure, and transfer trajectories are discretized into geometrically meaningful arcs based on curvature maxima. These arcs are clustered into groups of geometrically similar motions using hierarchical and density-based clustering algorithms. For periodic orbits, groups are constructed from continuous segments along a family with the same number of curvature maxima

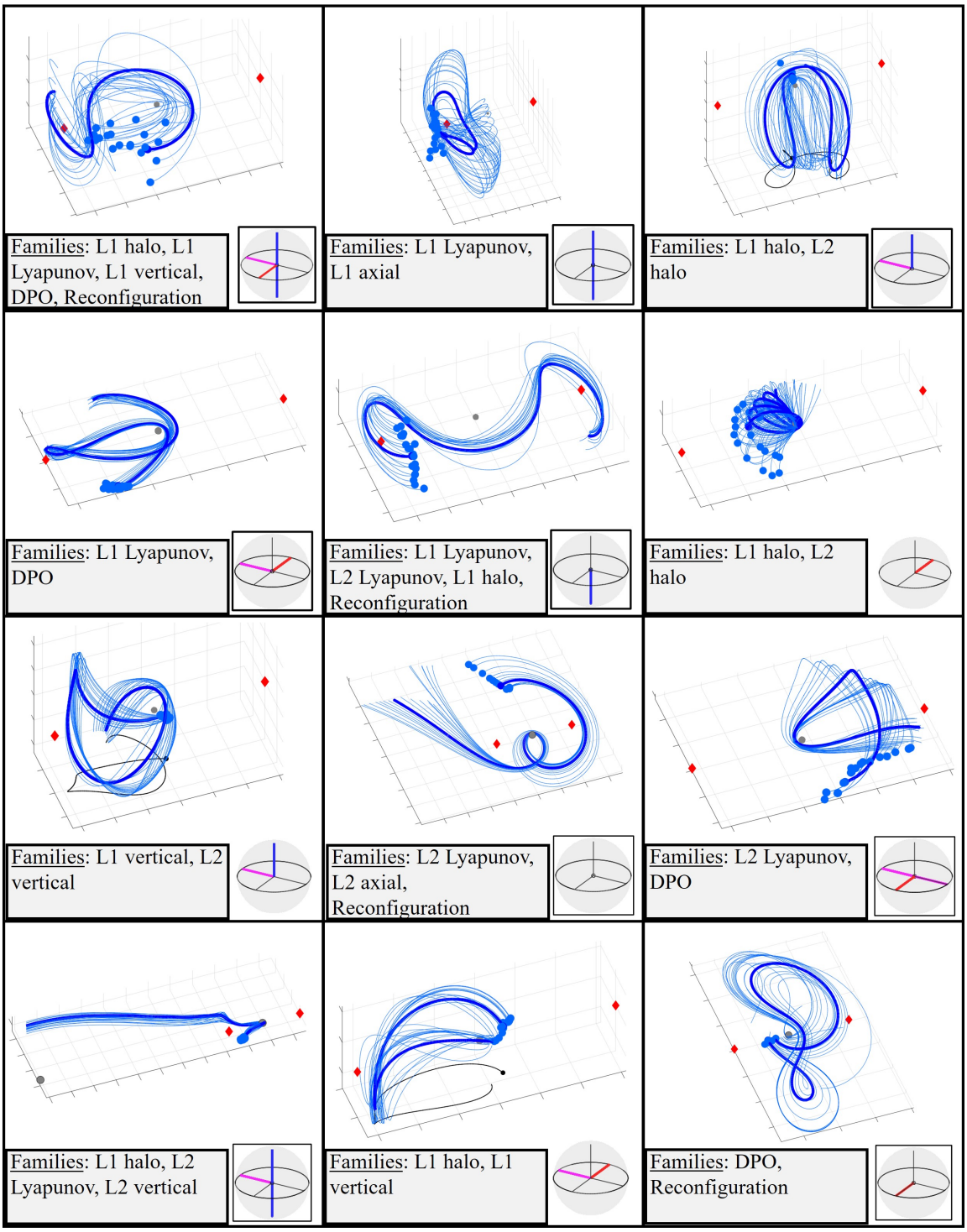

Figure 8. Aggregated behavioral motion primitives summarizing natural and thrust-enabled approaches into and departures from various bounded motions in the CR3BP.

along each orbit. Then, motion primitives are extracted from these groups. These primitives are labeled to identify the associated behavior and parameters producing motion in each group. Then, aggregation is used to merge similar groups from different behaviors together, merging their labels together in the process. Through this approach, an extensive and diverse library of behavioral motion primitives is generated with the goal of supporting spacecraft trajectory design and prediction in cislunar space.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-23-1-0235. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

REFERENCES

[1] A. Wolek and C. A. Woolsey, *Model-Based Path Planning*. Cham: Springer International Publishing, 2017, 10.1007/978-3-319-55372-6_9.

[2] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-Based Motion Planning for Nonlinear Systems with Symmetries," *IEEE Trans. Robot.*, Vol. 21, No. 6, 2005, pp. 1077–1091. DOI: https://doi.org/10.1109/TRO.2005.852260.

[3] A. Majumdar and R. Tedrake, "Funnel Libraries for Real-Time Robust Feedback Motion Planning," *CoRR*, Vol. abs/1601.04037, 2016.

[4] D. Clever, M. Harant, H. Koch, K. Mombaur, and D. Endres, "A Novel Approach for the Generation of Complex Humanoid Walking Sequences Based on a Combination of Optimal Control and Learning of Movement Primitives," *Robotics and Autonomous Systems*, Vol. 83, 2016, pp. 287–298, 10.1016/j.robot.2016.06.001.

[5] G. Habibi, N. Jaipuria, and J. P. How, "SILA: An Incremental Learning Approach for Pedestrian Trajectory Prediction," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4411–4421, 10.1109/CVPRW50498.2020.00520.

[6] O. Dermy, A. Paraschos, M. Ewerton, J. Peters, F. Charpillet, and S. Ivaldi, "Prediction of Intention during Interaction with iCub with Probabilistic Movement Primitives," *Frontiers in Robotics and AI*, Vol. 4, 2017, 10.3389/frobt.2017.00045.

[7] T. R. Smith and N. Bosanac, "Constructing Motion Primitive Sets to Summarize Periodic Orbit Families and Hyperbolic Invariant Manifolds in a Multi-Body System," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, No. 1, 2022, p. 7, 10.1007/s10569-022-10063-x.

[8] G. Miceli, N. Bosanac, and R. Karimi, "Generating The Trajectory Design Space For Neptunian System Exploration," *AAS/AIAA Astrodynamics Specialists Conference*, August 2024.

[9] G. E. Miceli and N. Bosanac, "Designing Neptunian System Tours via a Motion Primitive Approach," *AAS/AIAA Space Flight Mechanics Meeting*, 2025.

[10] T. R. Smith and N. Bosanac, "Motion Primitive Approach to Spacecraft Trajectory Design in a Multibody System," *The Journal of the Astronautical Sciences*, Vol. 70, No. 34, 2023, 10.1007/s40295-023-00395-7.

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

[12] B. A. Conway, *Spacecraft Trajectory Optimization*. Cambridge Aerospace Series, Cambridge University Press, 2010.

[13] K. Wardle, *Differential Geometry*. Mineola, NY: Dover Publications, Inc., 2008.

[14] N. Patrikalakis, T. Maekawa, and W. Cho, *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer, Berlin, Heidelberg, 2009. E-book.

[15] N. Bosanac, "Curvature Extrema Along Trajectories in the Circular Restricted Three-Body Problem," *AAS/AIAA Astrodynamics Specialists Conference*, 2024.

[16] J. Han, M. Kamber, and J. Pei, *Data Mining (Third Edition)*. The Morgan Kaufmann Series in Data Management Systems, Boston: Morgan Kaufmann, third edition ed., 2012.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, AAAI Press, 1996, p. 226–231.

[18] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," *Advances in Knowledge Discovery and Data Mining* (J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, 2013, pp. 160–172.

[19] C. Malzer and M. Baum, "A Hybrid Approach To Hierarchical Density-based Cluster Selection," *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2019, pp. 223–228.

[20] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical Density Based Clustering," *The Journal of Open Source Software*, Vol. 2, mar 2017, 10.21105/joss.00205.

[21] J. S. Parker and R. L. Anderson, *Low-energy Lunar Trajectory Design*. Hoboken, New Jersey: Wiley, 2014.

[22] W. Koon, M. Lo, J. Marsden, and S. Ross, *Dynamical Systems, The Three-Body Problem, and Space Mission Design*. New York, USA: Springer, 2011.

[23] MOOG, *ESPA USER'S GUIDE THE EELV SECONDARY PAYLOAD ADAPTER*.

[24] I. Elliott, C. Sullivan, N. Bosanac, J. R. Stuart, and F. Alibay, "Designing Low-Thrust Trajectories for a SmallSat Mission to Sun–Earth L5," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 10, 2020, pp. 1854–1864, 10.2514/1.G004993.

[25] Busek Co. Inc, *BHT-200*.

[26] N. Bosanac, "Data-Driven Summary of Motion in an Ephemeris Model of Cislunar Space," *AAS/AIAA Space Flight Mechanics Meeting*, Lihue, HI, 2025.

[27] N. Bosanac and M. Joyner, "Data-Driven Summary of Continuous Thrust Trajectories in a Low-Fidelity Model of Cislunar Space," *AAS/AIAA Astrodynamics Specialists Conference*, 2024.

[28] Y. Zheng and X. Zhou, *Computing with Spatial Trajectories*. Computing with Spatial Trajectories, Springer New York, 2011.