

# CONSTRUCTING A SET OF MOTION PRIMITIVES IN THE CIRCULAR RESTRICTED THREE-BODY PROBLEM VIA CLUSTERING

Thomas R. Smith\* and Natasha Bosanac†

To reduce the complexity of trajectory design in chaotic dynamical environments, data analysis techniques support the representation of a large and diverse solution space via a fundamental set of governing structures. Clustering is a data mining technique used to summarize a dataset by uncovering its underlying structures. In this paper, a variety of commonly used clustering algorithms are explored to construct a set of motion primitives that summarize a family of periodic orbits in the Circular Restricted Three-Body Problem (CR3BP). An overview of common clustering algorithms is provided and the motion primitive construction process for trajectories in a multi-body system is outlined. The impact of various clustering algorithms and feature vector definitions on the construction of motion primitives in the CR3BP is evaluated for the family of Distant Prograde Orbits (DPOs) in the Earth-Moon system.

## INTRODUCTION

Human exploration of the lunar surface through a cislunar waypoint, robotic exploration of asteroids and planetary systems, and the use of advanced space telescopes to study the evolution of the universe require the operation of spacecraft in chaotic environments.<sup>1-3</sup> For these missions, a rapid and intuitive trajectory design process enables the development of complex trajectories both during concept development and, potentially, post-launch. In multi-body systems, one current approach to rapid trajectory design involves generating a large database of solutions discretized along entire families of periodic and quasi-periodic orbits.<sup>4</sup> Specialized design tools then support the exploration and analysis of these complex multi-dimensional families to construct an initial guess for a trajectory.<sup>5</sup> However, searching over a large and complex design space may be infeasible in time-critical scenarios such as during rapid amplification of uncertainty in chaotic regimes, hazard response, maneuvering relative to nonresponsive agents, or unpredicted aborts. Furthermore, such an approach may impede rapid mission concept development. In these scenarios, trajectory design strategies need to be adaptable and robust in response to increasingly complex mission concepts, mission extensions, and real-time operations.

To enable intuitive and robust trajectory design, this paper focuses on constructing a set of motion primitives that reflect the underlying dynamics in a multi-body system. For instance, the Circular Restricted Three-Body Problem (CR3BP) is a useful autonomous dynamical model for preliminary analysis of the motion of a spacecraft near two larger bodies.<sup>6</sup> This model admits a variety

---

\*Graduate Research Assistant, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80309.

†Assistant Professor, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 431 UCB, Boulder, CO 80309.

of dynamical structures including: periodic orbits, quasi-periodic orbits, and invariant manifolds. These structures exist in continuous families. However, a family of solutions tends to exhibit a finite number of distinct geometries and other trajectory characteristics. Therefore, each family may be reduced to a finite set of representative fundamental solutions, i.e., a set of motion primitives. In this paper, a motion primitive is considered an average representation of a range of similar solutions. This concept of a motion primitive is similar to the use of motion primitives in robotic motion planning.<sup>7</sup> To identify suitable motion primitives, techniques from data analysis are useful.

Clustering algorithms have previously been used to simplify analysis and visualization of the structures present in nonlinear dynamical systems. For instance, spectral clustering has been employed by Hadjighasem, Karrasch, Teramoto, and Haller to identify coherent Lagrangian vortices from the flow of Lagrangian trajectories to develop a simplified description of the vortices in the underlying dynamical system.<sup>8</sup> In astrodynamics, the partitioning clustering algorithm,  $k$ -means, has been used by Nakhjiri and Villac to identify regions of stability from dynamical map structures and by Villac, Anderson, and Pini to group periodic orbit solutions into families based on apse locations and orbit period in the Augmented Hill’s Three-Body Problem (AH3BP).<sup>9,10</sup> In addition, Bosanac applied hierarchical density-based clustering methods to Poincaré maps in the CR3BP to group trajectories with similar geometries and produce a reduced representative dataset to reduce the complexity of visualization and facilitate analysis.<sup>11</sup> These applications of clustering algorithms demonstrate the value of a data mining approach in grouping trajectories of nonlinear and chaotic dynamical systems based on defining features – as well as their potential to be used more extensively in the trajectory design process.

The focus of this paper is to use clustering algorithms to construct a set of motion primitives that sufficiently represents a family of periodic orbits. The output of a clustering algorithm is highly dependent on the features used to represent each solution in the dataset as well as the type of clustering algorithm used (e.g. partition-based, hierarchical, density-based).<sup>12</sup> Several clustering algorithms – including  $k$ -means, spectral clustering, mean shift, agglomerative clustering, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), and affinity propagation – are applied to the problem of identifying a set of motion primitives that represents a family of periodic orbits in the CR3BP. The family of Distant Prograde Orbits (DPOs) in the Earth-Moon system exhibits multiple distinct geometries that cannot be clearly differentiated using analytical functions and, therefore, serves as a useful test case for this analysis. Each orbit within the family is characterized by several parameters that reflect its geometry and/or stability. Given range and amplitude discrepancies between features, a normalization scheme is also used to mitigate the impact of feature biases. Each clustering algorithm is then applied to the family of periodic orbits described by a specific feature parameterization and a set of motion primitives is constructed. The impact of each algorithm and feature vector on the resulting sets of motion primitives is compared. Each set of motion primitives constructed in this analysis provides a feasible summarization of the family of periodic orbits.

## **BACKGROUND: DYNAMICAL MODEL**

The CR3BP is a nonlinear dynamical model that is commonly used to approximate the motion of a spacecraft operating primarily under the gravitational influence of two bodies, such as the Earth and the Moon. The spacecraft, labeled  $P_3$ , is assumed to possess a negligible mass compared to the larger primary body,  $P_1$ , with constant mass  $M_1$  and the smaller primary body,  $P_2$ , with constant mass  $M_2$ . Consequently, the motion of  $P_3$  does not impact the motion of  $P_1$  and  $P_2$ .

Additionally, both primary bodies are modeled as point masses. As a result, the center of mass, or barycenter, of the system lies between  $P_1$  and  $P_2$  and the two primary bodies act as an isolated two-body problem.<sup>6</sup> Then,  $P_1$  and  $P_2$  are assumed to follow circular orbits around their mutual barycenter. Through this assumption, an explicit dependence on time is removed from the system by describing the motion of  $P_3$  in a rotating reference frame: the origin of the reference frame is at the barycenter of the system, the  $\hat{x}$ -axis lies along the line between  $P_1$  and  $P_2$ , the  $\hat{z}$ -axis is aligned with the angular momentum of the system, and the  $\hat{y}$ -axis completes the right-handed orthogonal coordinate frame. The quantities describing the motion of  $P_3$  in this rotating reference frame are nondimensionalized using the characteristic parameters  $l^*$ ,  $m^*$ , and  $t^*$ . Specifically,  $l^*$  is set equal to the distance between  $P_1$  and  $P_2$ ,  $m^*$  corresponds to the total mass of the system, and  $t^*$  is defined such that the period of the primaries around their mutual barycenter is equal to  $2\pi$ . The state of  $P_3$  is then written in nondimensional position and velocity coordinates  $x, y, z, \dot{x}, \dot{y}$ , and  $\dot{z}$  relative to the barycenter of the system in the rotating frame. Given these assumptions and definitions, the nondimensional equations of motion for  $P_3$  in the rotating reference frame and in the CR3BP are written as:

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

where  $\mu = \frac{M_2}{M_1+M_2}$  is the mass ratio of the system,  $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$  is the distance between  $P_3$  and  $P_1$ , and  $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$  is the distance between  $P_3$  and  $P_2$ . In this nonlinear, chaotic system, an integral of motion, commonly labeled the Jacobi constant, is conserved for natural trajectories and is equal to:

$$C_J = (x^2 + y^2) + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} - \dot{x}^2 - \dot{y}^2 - \dot{z}^2\tag{2}$$

At a single value of  $C_J$ , a variety of natural dynamical structures such as periodic orbits, quasi-periodic orbits, and invariant manifolds exist throughout the phase space. These natural structures exist in one- or multi-dimensional families and can be computed numerically via dynamical systems theory. However, analysis and visualization in the CR3BP is complex due to the large variety of diverse solutions that exist throughout the system. To simplify the solution space, data analysis techniques such as clustering are invaluable.<sup>6,13</sup>

## BACKGROUND: CLUSTERING ALGORITHMS

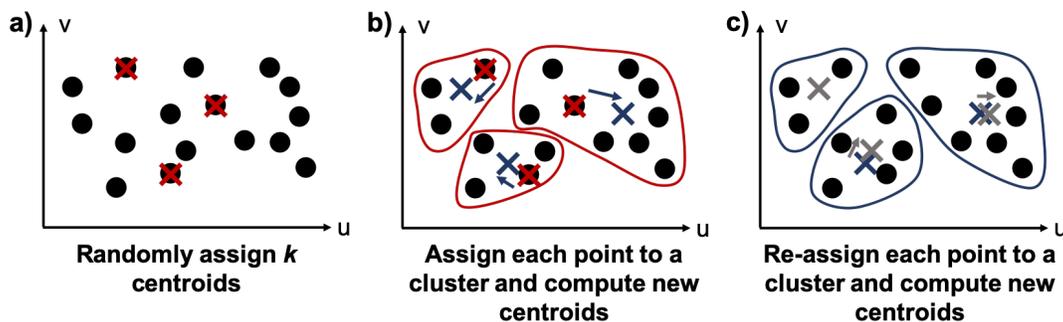
A clustering algorithm divides a dataset into separate groupings, or clusters, such that data in the same cluster are similar and data in different clusters are dissimilar.<sup>12</sup> To cluster the members of a dataset, a list of properties or features that describe each object, i.e., a feature vector denoted as  $\vec{f}$ , must be defined. Selecting a feature vector is object-specific because the available features depend on the type of objects in the dataset, but it is also application-specific because the features should be selected such that the clustering algorithm develops a summarization of the dataset useful for a desired application. However, the results of a clustering algorithm are also dependent on the method

of clustering. Common types of clustering include partitioning, hierarchical, and density-based methods. Partitioning methods, such as  $k$ -means, partition a dataset of  $n$  objects into  $k$  clusters, where  $k \leq n$ , by assigning cluster centers and iteratively placing each object into a cluster using a distance-based metric. Hierarchical methods utilize either a “bottom-up” (agglomerative) or “top-down” (divisive) approach to develop a hierarchy tree structure of clusters in which the levels of the tree correspond to inter-cluster similarities that are typically determined by either a distance or density-based metric. Finally, density-based methods form clusters by defining a neighborhood around each object in the dataset and building clusters based on the densities of the neighborhoods. Each of these methods require the selection of input parameters and can produce fundamentally different clustering results. Thus, an analysis that evaluates the clustering results for multiple algorithms applied to the same dataset is warranted. In this section, an overview is provided for the clustering algorithms that are leveraged in this analysis.

### ***K*-Means**

The  $k$ -means algorithm is a partition-based clustering method that groups members of a dataset based on the distance of each member from  $k$  centroids. The algorithm requires the number of clusters,  $k$ , as an input; then, the centroids are initialized by randomly selecting  $k$  members of the dataset.<sup>12</sup> The clusters are formed by assigning each object to the closest centroid evaluated using the  $l^2$ -norm. The centroid of each cluster is then recomputed and the objects are reassigned to the new centroids to form new clusters. These steps are depicted conceptually in Figures 1a), 1b), and 1c) respectively on a two-dimensional dataset for  $k = 3$ . This process is repeated iteratively with the goal of minimizing the sum of the squared Euclidean distances between the centroid of each cluster and the objects assigned to each cluster. The algorithm terminates either when the clusters remain unchanged from one iteration to the next or a maximum number of iterations is exceeded.  $K$ -means will always converge but the algorithm may only find a local minimum dependent on the selection of the initial centroids.<sup>14</sup> To increase the stability and robustness of using  $k$ -means clustering, a common technique is to run the algorithm multiple times with different initial centroids and select the clustering results with the lowest inertia. The inertia,  $E$ , is a metric that can be used to evaluate the performance of  $k$ -means and is defined as:

$$E = \sum_{i=1}^k \sum_{\vec{f} \in C_i} \|\vec{f} - \vec{\mu}_i\|^2 \quad (3)$$



**Figure 1.** Example of  $k$ -means performed on a two-dimensional dataset where  $k = 3$  and the feature vector of each data point contains its  $u$  and  $v$  coordinates.

where  $C_i$  is the  $i^{th}$  cluster,  $\vec{f}$  is the feature vector of a member in the  $i^{th}$  cluster, and  $\vec{\mu}_i$  is the centroid of the  $i^{th}$  cluster. The primary limitation of  $k$ -means is selecting the number of clusters, but the inertia can be evaluated as a function of  $k$  and used to provide a heuristic metric to appropriately estimate the value of  $k$  without a priori knowledge of the dataset.<sup>12,14</sup>

## Spectral Clustering

Spectral clustering uses a spectral decomposition of a dataset to reduce the dimensionality and perform  $k$ -means in a lower dimensional subspace.<sup>15</sup> This dimensionality reduction is particularly useful for clustering large multi-dimensional datasets. Initially, a similarity matrix,  $A$ , is constructed from the dataset to numerically represent the similarities between each object. A radial basis function kernel is typically used to compute this similarity metric and the  $(i, j)$  element of the  $A$  matrix is defined as:

$$A_{i,j} = e^{-\frac{\|\vec{f}_i - \vec{f}_j\|^2}{2\sigma^2}} \quad (4)$$

where  $\vec{f}_i$  is the feature vector of the  $i^{th}$  member in the dataset,  $\vec{f}_j$  is the feature vector of the  $j^{th}$  member in the dataset, and  $\sigma$  is a scaling parameter that controls the bandwidth of the kernel. The value of  $\sigma$  is often set equal to  $\frac{1}{\sqrt{2}}$  as a default but, otherwise, needs to be specified.<sup>14</sup> Using the radial basis function kernel,  $A$  represents a fully-connected similarity graph. Computed from  $A$ , the degree matrix,  $D$ , is a diagonal matrix defined such that the  $(i, i)$  element is equal to:

$$D_{i,i} = \sum_{j=1}^n A(i, j) \quad (5)$$

where each diagonal entry in  $D$  corresponds to the sum of the similarities between the  $i^{th}$  member in the dataset and each of the other members in the dataset. The similarity and degree matrices are then used to compute the symmetric normalized graph Laplacian matrix,  $L$ , as:

$$L = I_{n \times n} - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (6)$$

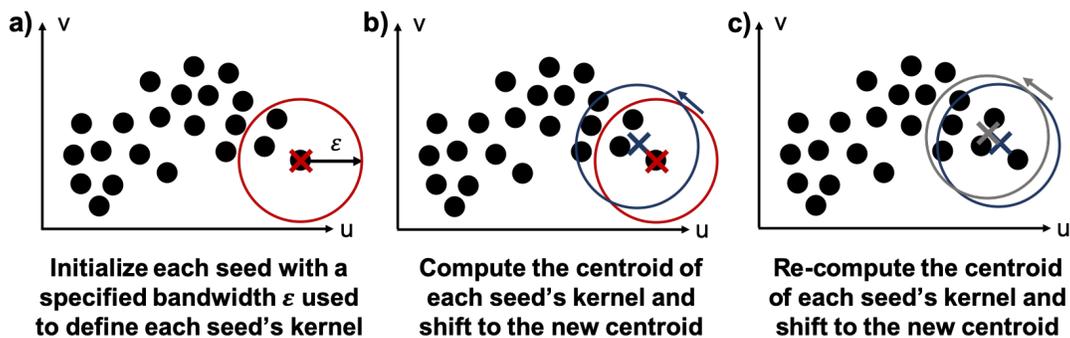
where  $I_{n \times n}$  is the  $n \times n$  identity matrix. The  $k$  largest eigenvectors of  $L$  are then used to construct a  $n \times k$  matrix,  $X$ , in which each column corresponds to an eigenvector of  $L$ .<sup>14-16</sup> This matrix projects each original feature vector in the dataset from  $\mathbb{R}^m$ , where  $m$  is the number of features in each original feature vector, to  $\mathbb{R}^k$ , where  $k$  is the number of components in each reduced feature vector.  $K$ -means clustering is then applied to the reduced dataset stored in  $X$  and the clustering labels assigned to each solution in  $X$  are mapped back to the original dataset. This algorithm, therefore, requires specification of  $k$ , the number of clusters. Consequently, spectral clustering also relies on a priori knowledge of the dataset to select an appropriate number of clusters. However, spectral clustering performs  $k$ -means in a lower dimensional subspace and, therefore, increases the efficiency of the algorithm. An advantage of spectral clustering is the use of the graph Laplacian matrix to represent the dataset because its eigenvalues and eigenvectors can provide insight into the natural structure of clusters found in the dataset.

## Mean Shift

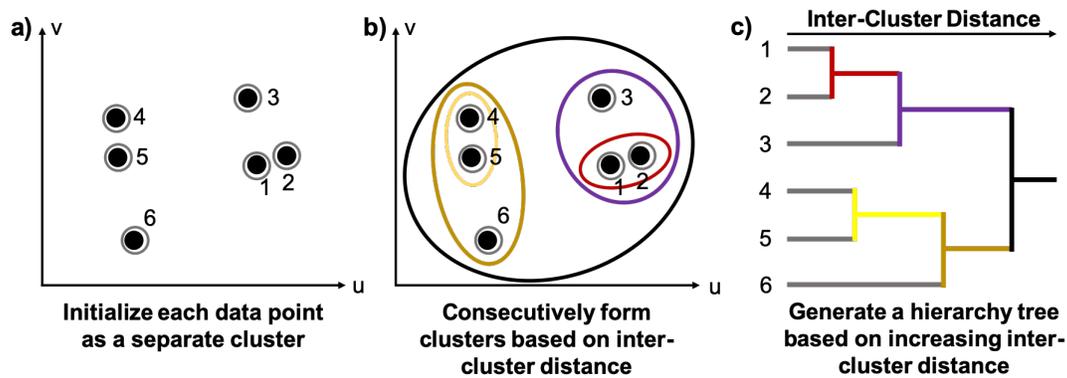
The mean shift algorithm is a partition-based method; however, unlike  $k$ -means, the algorithm considers a dataset as a discrete sampling of a density function from which it locates the modes of the function using an iterative gradient-based approach.<sup>14,17</sup> Initially, all objects in the set are considered seeds, i.e., potential modes, with a specified bandwidth,  $\varepsilon$ . The bandwidth is used to define the size of the kernel, or neighborhood, of each seed. Various kernels can be used, but a flat kernel is used for the example shown in Figure 2 where mean shift is performed on a two-dimensional dataset.<sup>17</sup> In this example, the kernel of each seed is simply a circle with radius  $\varepsilon$  centered at the seed, as depicted conceptually in Figure 2a). The centroid of the points within the kernel of each seed is then computed and each seed is shifted to its new centroid, as depicted in Figure 2b) for a single seed. This process is repeated and each seed continues to iteratively shift in its mean shift direction, as depicted in Figure 2c), until convergence. The mean shift direction is the vector directed from each seed to the centroid of its kernel. Convergence occurs when the mean shift of each seed is less than a specified tolerance. Additionally, seeds that converge to the same mode of the density function are merged to define a single mode.<sup>14,18</sup> Consequently, each mode corresponds to a cluster and each member of the dataset is assigned to the closest mode using the  $l^2$ -norm as a measure of similarity. In contrast to  $k$ -means, the mean shift algorithm does not require the number of clusters to be specified a priori. Rather, the number of clusters is recovered naturally in this mode seeking approach. The results are, however, highly dependent on the bandwidth used to define the size of the kernel for each seed. An appropriate bandwidth may be estimated based on the distances between all of the objects in the dataset using a quantile,  $q$ . The quantile determines a bandwidth such that a certain percentage of the distances between each of the objects are less than the value of the bandwidth. For example,  $q = 0.5$  results in the bandwidth being selected as the median distance between all of the objects because 50% of the distances are less than the median distance. The quantile input provides an intuitive parameter to estimate the bandwidth, but the mean shift algorithm still requires some a priori knowledge of the dataset to select a proper quantile.

## Agglomerative Clustering

Agglomerative clustering uses a bottom-up approach to generate a hierarchical decomposition of a dataset. This hierarchical decomposition is stored in a tree with each node representing a cluster.



**Figure 2.** Example of mean shift performed on a two-dimensional dataset where the feature vector of each data point contains its  $u$  and  $v$  coordinates. For clarity, the steps presented are only shown for a single seed, but in the algorithm these steps are executed simultaneously for all of the seeds. [Adapted from Ukrainitz and Sarel]<sup>19</sup>



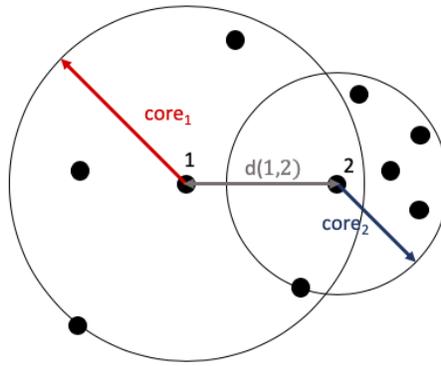
**Figure 3.** Example of agglomerative clustering performed on a two-dimensional dataset where the feature vector of each data point contains its  $u$  and  $v$  coordinates.

For a dataset composed of  $n$  members, the tree has  $n$  leaves and each leaf corresponds to a separate cluster.<sup>12</sup> The agglomerative clustering process is depicted conceptually in Figure 3 for a sample two-dimensional dataset. Initially, all of the objects are considered separate clusters as depicted in Figure 3a). At each step of the algorithm, the distances between all of the current clusters are computed and the pair of clusters with the smallest inter-cluster distance (the distance between two clusters as defined using a suitable metric) is combined into a new cluster. This process continues until all of the objects are grouped into a single cluster as portrayed in Figure 3b). The tree of clusters generated using agglomerative clustering is often visualized as a dendrogram, as displayed in Figure 3c). In this figure, all of the objects are initially allocated to separate clusters (gray) and combine into new clusters at various inter-cluster distances (red, yellow, purple, and orange) until, finally, all of the clusters are merged together (black). A desired number of clusters is specified as an input to the algorithm and the tree is cut at an inter-cluster distance that results in the specified number of clusters.

The dendrogram produced in agglomerative clustering provides a useful visualization tool to select an appropriate number of clusters, but selecting the metric used to compute inter-cluster distance, known as the linkage type, is often nonintuitive and changes the underlying tree structure. Common linkage types include single, complete, average, and Ward linkage.<sup>12,14</sup> Single linkage measures inter-cluster distance as the distance between the closest two objects in the two clusters while complete linkage uses the farthest two objects. Average linkage measures inter-cluster distance as the average of the distances between all of the objects in the two clusters while ward linkage computes the variance of the combined cluster if the two clusters were to be merged. Furthermore, the Euclidean distance is used in Ward linkage but any similarity metric can be used in the other linkage types. The selected linkage type impacts the results of the algorithm, producing distinctly different tree structures and, therefore, different clustering results for the same dataset. Ward linkage is commonly used because it typically produces even cluster sizes and uses a variance-based metric similar to  $k$ -means. However, selecting the appropriate linkage type relies on a priori knowledge of the dataset.

### **Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)**

HDBSCAN is a hierarchical density-based method that enables the construction of clusters with varying densities and arbitrary shapes.<sup>20</sup> In this algorithm, a minimum cluster size,  $minClusterSize$ ,

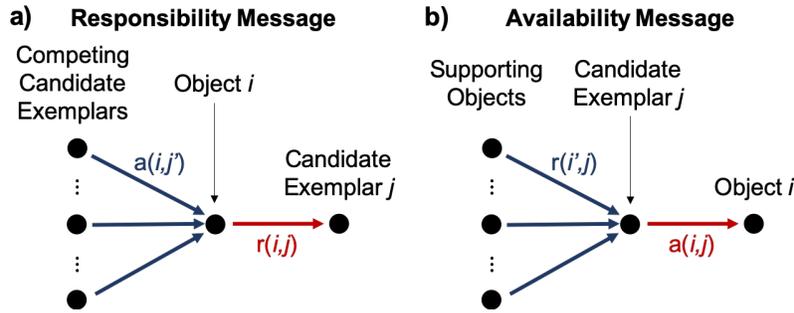


**Figure 4.** Example of the core distance metric used in HDBSCAN for two objects when  $minPts = 6$ . [Adapted from McInnes, Healy, and Astels]<sup>21</sup>

and a minimum neighborhood size,  $minPts$ , are specified as inputs. The  $minClusterSize$  is often specified based on the size of the dataset but selecting  $minPts$  is not as intuitive. The  $minPts$  is used to compute the core distance for each object, defined as the distance to its  $i^{th}$  nearest neighbor where  $i = minPts - 1$  as depicted conceptually in Figure 4 for two objects when  $minPts = 6$ . The mutual reachability distance between two points is then defined as the maximum value among the core distance of the first object, the core distance of the second object, and the distance between the two objects. This distance is computed between all objects in the dataset and used to construct a minimum spanning tree where each vertex is an object and the edge between two objects has a weight equal to their mutual reachability distance. From the minimum spanning tree, an agglomerative approach is used to develop a hierarchy of clusters based on increasing mutual reachability distance. The hierarchy is then condensed by evaluating the stability of each cluster throughout the hierarchy. If a cluster splits into two separate clusters and one of the new clusters does not meet the specified  $minClusterSize$  requirement, then this cluster split is disregarded. The objects that split off are removed from the persisting cluster. This approach distinguishes between new cluster generation and cluster size reduction. From the condensed tree, the most stable clusters are identified and used to represent the dataset. HDBSCAN leverages both hierarchical and density-based clustering to improve upon density-based methods by simplifying the user input parameters and enabling the construction of a number of clusters that is not known a priori and with varying densities. However, the selection of  $minPts$  is often an iterative process.

## Affinity Propagation

Affinity propagation is an exemplar-based clustering algorithm in which numeric messages are sent between objects to find exemplars that sufficiently represent a dataset.<sup>22</sup> The number of exemplars is equivalent to the number of clusters because each exemplar is the most representative solution for a given cluster. There are two types of numeric messages transmitted between objects: responsibility messages and availability messages. A responsibility message,  $r(i, j)$ , is sent from an object  $i$  to a candidate exemplar  $j$  to indicate the similarity of these two objects relative to the other competing exemplars for object  $i$ . On the other hand, an availability message,  $a(i, j)$ , is sent from a candidate exemplar  $j$  to an object  $i$  to indicate the support object  $j$  has from other objects to be an exemplar. A conceptual overview of a responsibility and an availability message is depicted in Figures 5a) and 5b), respectively. Initially, all responsibility and availability messages are set equal



**Figure 5.** Conceptual depiction of a responsibility message and an availability message sent in affinity propagation where object  $j'$  is a competing candidate exemplar and object  $i'$  is a supporting object. [Adapted from Frey and Dueck]<sup>22</sup>

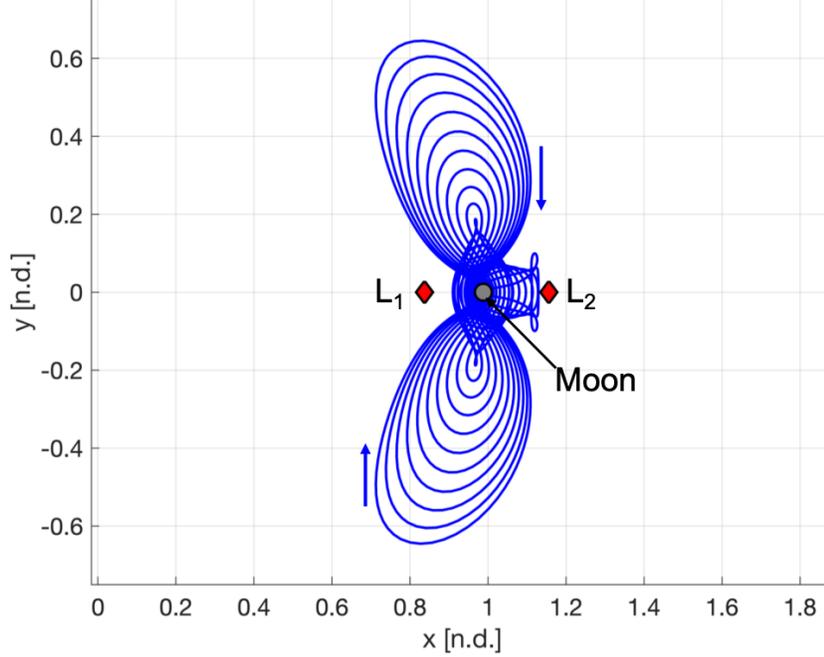
to zero. Additionally, sample preferences can be set to bias the likelihood that certain objects are selected as exemplars. If no a priori knowledge of the dataset is available, all sample preferences are set equal to the median of the similarity values in the dataset. An additional input parameter is a damping factor ranging from 0.5 to 1 used to avoid numerical oscillations when updating messages.<sup>14,22</sup> During each iteration of the algorithm, the responsibility messages and availability messages of each object are updated. Then, the exemplar for each object is determined by finding the object that maximizes the sum of its responsibility and availability messages. If the object that maximizes this sum is itself, then the point is classified as an exemplar. These updates are repeated until the clustering results remain unchanged for a specified number of iterations or a maximum number of iterations is exceeded. Once the algorithm converges, each exemplar and its supporting objects form a cluster. For large datasets, affinity propagation has a high computational time and memory storage complexity. However, the outputs of affinity propagation are particularly useful for extracting motion primitives because each exemplar is the most representative solution of a cluster.

## TECHNICAL APPROACH: MOTION PRIMITIVE CONSTRUCTION

Clustering algorithms are leveraged in this analysis to construct motion primitives that represent the finite set of characteristics exhibited by a family of periodic orbits. In the CR3BP, a family of periodic orbits corresponds to a continuous set of solutions that often exhibit a finite number of distinct geometries and stability descriptions. However, there is no clear analytical expression for separating or grouping these solutions. For example, the family of Distant Prograde Orbits (DPOs) encircling the Moon in the Earth-Moon system is composed of members that exhibit multiple geometries. Figure 6 displays a subset of the DPO family computed in the Earth-Moon CR3BP using a multiple shooting corrections algorithm and pseudo-arclength continuation.<sup>13,23</sup> These periodic orbits are plotted in the rotating frame and the arrows indicate direction of motion. The full dataset computed in this example consists of 400 DPOs sampled along the family. Using this dataset, this section outlines the selection of a feature vector, followed by the definition of a motion primitive.

### Feature Vector Selection

The results of a clustering algorithm depend on the parameters, i.e. features, used to represent each member of the dataset. A finite dataset contains  $n$  objects where each object is defined by a  $m$ -dimensional feature vector. Common trajectory parameters in the CR3BP that may be used as features include, but are not limited to, Jacobi constant, orbit period, positions, velocities, stability



**Figure 6. Subset of the DPO family in the Earth-Moon system, displayed in the rotating frame.**

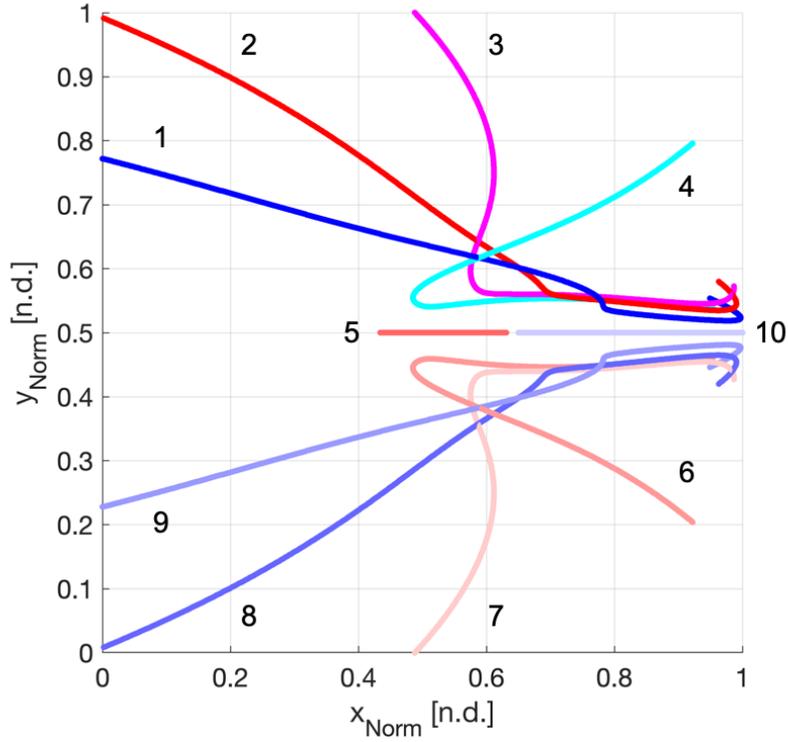
indices, apses locations, and number of apses. In this paper, three fundamental feature vectors are defined: a geometric, a parametric, and a geometric and stability feature vector.

The geometry of an orbit can be represented by a variety of different feature vectors. One representation of the geometry of a planar orbit involves using the normalized in-plane position components of states that are equally spaced in time along the orbit and can be defined as:

$$\vec{f} = \left[ \frac{(x_1 - 1 + \mu) - x_{min}}{x_{max} - x_{min}} \quad \frac{y_1 - y_{min}}{y_{max} - y_{min}} \quad \dots \quad \frac{(x_l - 1 + \mu) - x_{min}}{x_{max} - x_{min}} \quad \frac{y_l - y_{min}}{y_{max} - y_{min}} \right] \quad (7)$$

where  $l$  is the number of discrete positions sampled along the orbit and the  $x$ -positions are defined relative to the Moon because this family of orbits encircles the Moon. Additionally, all  $x$  and  $y$  positions are normalized between 0 and 1 using the minimum and maximum  $x$  and  $y$  values, respectively, along all members of the family. Applying this normalization scheme to the feature vector of each orbit ensures that all features are defined within the same range. Therefore, the same emphasis is placed on each feature and the clustering results are not biased towards a particular component. Figure 7 displays the geometric feature space produced for the family of DPOs when  $l = 10$  using the feature vector defined in Eq. 7. The 10 normalized position states in the feature vector of each orbit are denoted and distinguished by color. For example, the first normalized position state for each orbit in the family is denoted in dark blue. Figure 7 depicts the variations in each of the 10 normalized position states along the family of DPOs.

In addition to geometric features, various parametric features, such as energy and periodicity, may be used to characterize an orbit. A parametric representation of an orbit in the CR3BP is defined



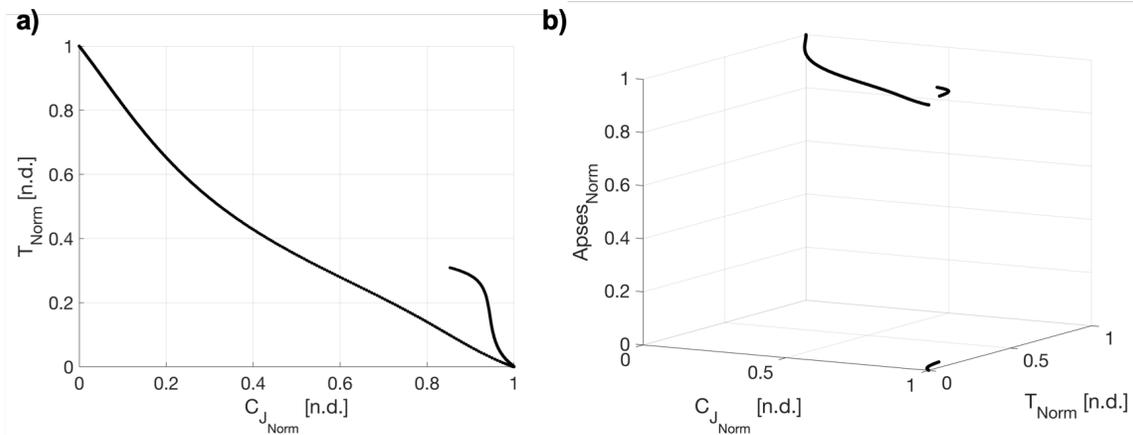
**Figure 7. Representation of the geometric feature space produced for the family of DPOs in the Earth-Moon system when  $l = 10$  using the feature vector defined in Eq. 7, containing only  $x$  and  $y$  positions, normalized to lie in the range  $[0,1]$ .**

using the following feature vector:

$$\vec{f} = \left[ \frac{C_J - C_{J_{min}}}{C_{J_{max}} - C_{J_{min}}}, \frac{T - T_{min}}{T_{max} - T_{min}}, \frac{Apses - Apses_{min}}{Apses_{max} - Apses_{min}} \right] \quad (8)$$

where  $C_J$  is the Jacobi constant of the orbit,  $T$  is the period of the orbit, and  $Apses$  is the total number of apses relative to a primary body throughout the orbit. This feature vector compresses the geometry of an orbit into a significantly lower dimensional representation as well as its energy and periodicity. For each orbit in the family of DPOs,  $Apses$  is equal to the total number of apses relative to the Moon throughout one period. Similar to the feature vector defined in Eq. 7, a normalization scheme is applied to avoid feature biases. The  $C_J$ ,  $T$ , and  $Apses$  of each orbit are normalized from 0 to 1 using the minimum and maximum values of  $C_J$ ,  $T$ , and  $Apses$  along the entire family of orbits. The parametric feature space describing the family of DPOs using the feature vector defined in Eq. 8 is visualized in Figures 8a) and 8b). The normalized period of each orbit in the family is depicted as a function of its normalized Jacobi constant in Figure 8a) while both of these parameters are plotted with respect to the normalized number of apses relative to the Moon for each orbit throughout the family in Figure 8b).

Similar to selecting features that represent the geometry of an orbit, features describing the stability of an orbit may also be included in a feature vector. The stability indices,  $s_1$  and  $s_2$ , of a periodic orbit in the CR3BP are used to evaluate the local stability of an orbit and are computed



**Figure 8. Representations of the parametric feature space produced for the family of DPOs in the Earth-Moon system using the feature vector defined in Eq. 8: a) orbit period as a function of Jacobi constant and b) the number of apses along each orbit as a function of Jacobi constant and orbit period normalized to lie in the range [0,1].**

from the eigenvalues of its monodromy matrix.<sup>13</sup> If a stability index is between -2 and 2 then it corresponds to an oscillatory mode. If  $s$  is not within this range, then it is associated with unstable and stable modes. Using this definition of stability,  $s_1$  and  $s_2$  are redefined as binary variables to provide a distinct separation between oscillatory and unstable/stable indices. If a stability index is in the range  $(-2, 2)$ , then it is reassigned to a value of 1; otherwise, it is reassigned to a value of 0. Defining the stability indices in this manner offers a clear distinction between different stability modes throughout the family of orbits. Using this modified definition of  $s_1$  and  $s_2$ , a feature vector representing both the geometry and local stability of a planar orbit is defined as:

$$\vec{f} = \left[ \frac{(x_1 - 1 + \mu) - x_{min}}{x_{max} - x_{min}} \quad \frac{y_1 - y_{min}}{y_{max} - y_{min}} \quad \dots \quad \frac{(x_l - 1 + \mu) - x_{min}}{x_{max} - x_{min}} \quad \frac{y_l - y_{min}}{y_{max} - y_{min}} \quad s_1 \quad s_2 \right] \quad (9)$$

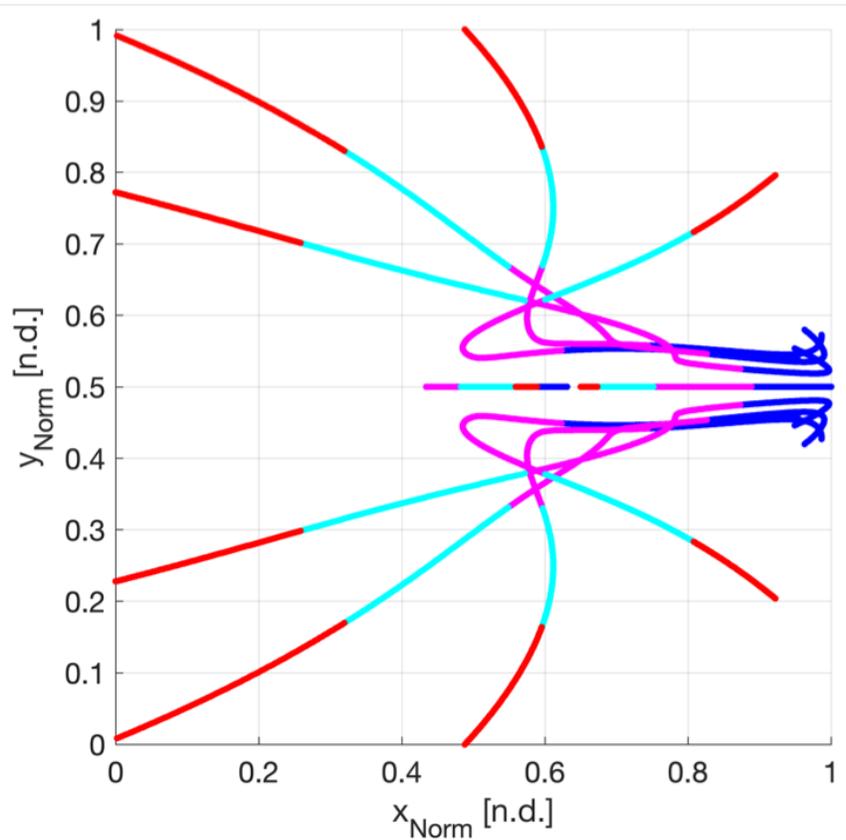
where  $\vec{f}$  is the vector defined in Eq. 7 augmented by  $s_1$  and  $s_2$ . The eigenvectors corresponding to  $s_1$  for the family of DPOs are directed only in the  $x$ - $y$  plane and, therefore,  $s_1$  corresponds to the in-plane stability of each DPO while  $s_2$  corresponds to the out-of-plane stability of each DPO. Since the family of DPOs are planar orbits, their stability indices offer a convenient description of in-plane and out-of-plane local stability. However, this is not always the case. The stability indices of three-dimensional periodic orbits do not provide this clear distinction between in-plane and out-of-plane stability, which is important to note when using this feature vector to cluster a non-planar family of orbits. However, the feature vector defined in Eq. 9 is simply one of many possible summarizations of an orbit for use in a clustering algorithm.

### Extracting Motion Primitives

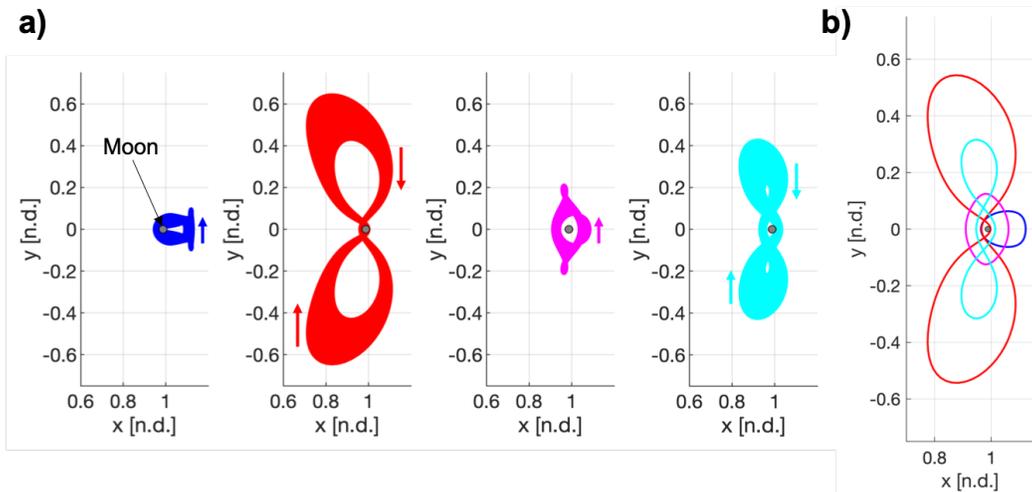
Once the feature vectors for a dataset have been computed, the data may be clustered and a set of motion primitives is extracted from the clustering results. In this analysis, all clustering results for  $k$ -means, spectral clustering, mean shift, agglomerative clustering, HDBSCAN, and affinity propagation are produced using either the Scikit-Learn v0.20.3 clustering module or the HDBSCAN clustering module available in Python.<sup>14,21</sup>  $K$ -means, mean shift, and affinity propagation all output

the feature vector corresponding to the center of each cluster, but the remaining algorithms do not define cluster centers. In the case where the cluster centers are not explicitly defined, the center of each cluster is identified by computing its centroid. However, the centroid of each cluster is not guaranteed to be a member of the dataset and therefore the centroids of a set of clusters cannot be used to define the motion primitives. Consequently, the motion primitive for a given cluster is extracted by finding the member of the cluster that is closest to its centroid using the appropriate similarity metric for the corresponding algorithm.<sup>12</sup>

The motion primitive construction process is demonstrated using the clustering results produced by  $k$ -means given a geometric representation of the DPO family. For example, the family of DPOs in the Earth-Moon system is partitioned into four clusters using  $k$ -means ( $k = 4$ ) where each orbit is represented by the geometry feature vector defined in Eq. 7, containing only  $x$  and  $y$  positions and sampled at 10 distinct locations along each orbit. The partitioned feature space is summarized in Figure 9, where the features of each orbit are colored based on their assigned cluster. The resulting clusters are visualized in Figure 10a) while the motion primitives extracted from these clusters are plotted in Figure 10b) in configuration space, capturing the distinct geometries present in the DPO family. This example demonstrates the motion primitive construction process used to summarize a family of periodic orbits in the CR3BP via a representative set of motion primitives.



**Figure 9.** Feature space representation of the  $k$ -means clustering results for the DPO family in the Earth-Moon system where  $k = 4$ . Each orbit is represented by the orbit geometry feature vector defined in Eq. 7 containing only  $x$  and  $y$  positions and  $l = 10$ .



**Figure 10. a) Clustering results for  $k$ -means ( $k = 4$ ) applied to the family of DPOs in the Earth-Moon system using the orbit geometry feature vector defined in Eq. 7 containing only  $x$  and  $y$  positions and  $l = 10$ . b) Motion primitives extracted from the family of DPOs in the Earth-Moon system using these  $k$ -means clustering results.**

## RESULTS: CLUSTERING APPLIED TO PERIODIC ORBITS IN THE CR3BP

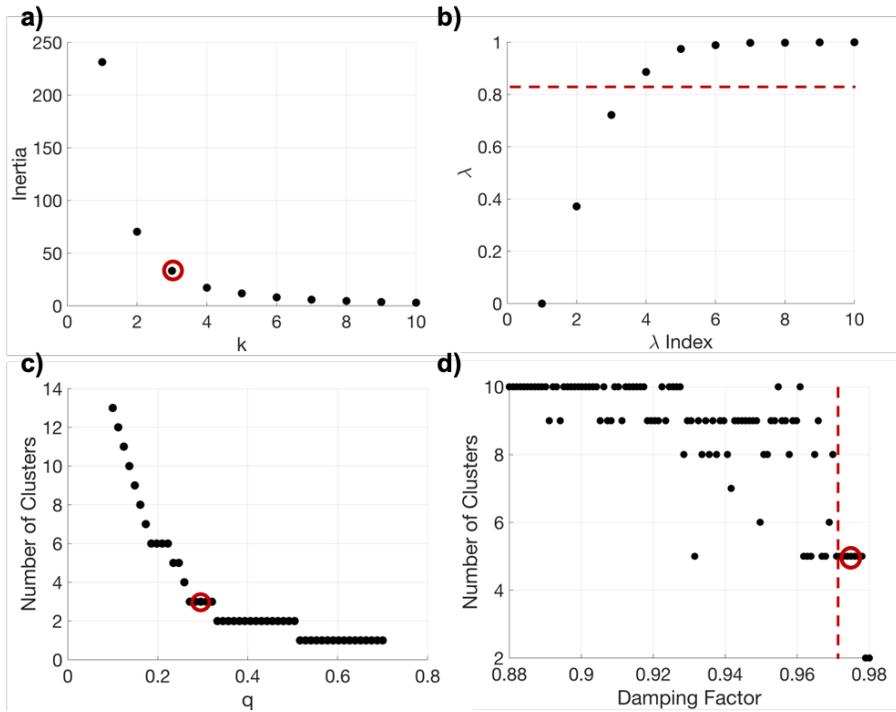
Motion primitives are constructed via various feature vectors and clustering algorithms to understand their impacts on the resulting primitives. The utilization of a geometric feature space and  $k$ -means in the previous section to demonstrate the motion primitive construction process is simply one of many choices for representing and clustering a family of periodic orbits in the CR3BP. A distinctly different set of motion primitives could be extracted from the family of DPOs if a different feature space and clustering algorithm are used. Furthermore, all clustering algorithms require user inputs and, therefore, typically require a priori knowledge of the dataset. However, various heuristics are often used to select appropriate input parameters for each clustering algorithm. Leveraging input parameter selection heuristics, this analysis explores the effect of applying  $k$ -means, spectral clustering, mean shift, agglomerative clustering, HDBSCAN, and affinity propagation to the family of DPOs in the Earth-Moon system using three different feature space representations.

### Geometric Feature Space

A geometric feature space is defined for the DPO family using the feature vector defined in Eq. 7 containing only  $x$  and  $y$  positions for  $l = 10$ , resulting in a 20-dimensional feature space comprising normalized position coordinates measured relative to the Moon. Using this representation of the DPO family, the input parameters for each algorithm are selected heuristically as follows:

- *K-Means*  $K$ -means is applied to the DPO family for a range of  $k$  values from 1 to 10. The inertia is computed for each set of clusters and the results are summarized in Figure 11a). Using the elbow method heuristic, the turning point in the curve occurs at approximately  $k = 3$ ; thus, this value of  $k$  is selected.<sup>12</sup>

- *Spectral Clustering* Figure 11b) depicts the first 10 eigenvalues of  $L$  computed from the DPO family. The number of zero eigenvalues is equal to the number of connected components, or distinct clusters, in the graph represented by  $L$ . When distinct clusters are not present, an eigengap heuristic



**Figure 11. Input parameter evaluation for  $k$ -means a), spectral clustering b), mean shift c), and affinity propagation d) applied to the family of DPOs.**

can be used to identify large changes in magnitude between two subsequent eigenvalues.<sup>15</sup> The number of clusters is estimated as the index of the eigenvalue just before the gap occurs. However, this metric becomes rather ambiguous when there is only one eigenvalue equal to zero, such as in Figure 11b). From the profile of eigenvalues,  $k = 3$  is selected because the gap between the third and fourth eigenvalues indicates a change in the trend of  $L$ 's eigenvalues.

- *Mean Shift* Figure 11c) displays the number of clusters produced by applying mean shift to the DPO family for a range of quantiles from 0.1 to 0.7. Based on this parameter search,  $q = 0.296$  is selected because, similar to the elbow method heuristic used in the  $k$ -means analysis, it is in the range of  $q$  values that appear to be at the bend in the curve as  $q$  is increased.

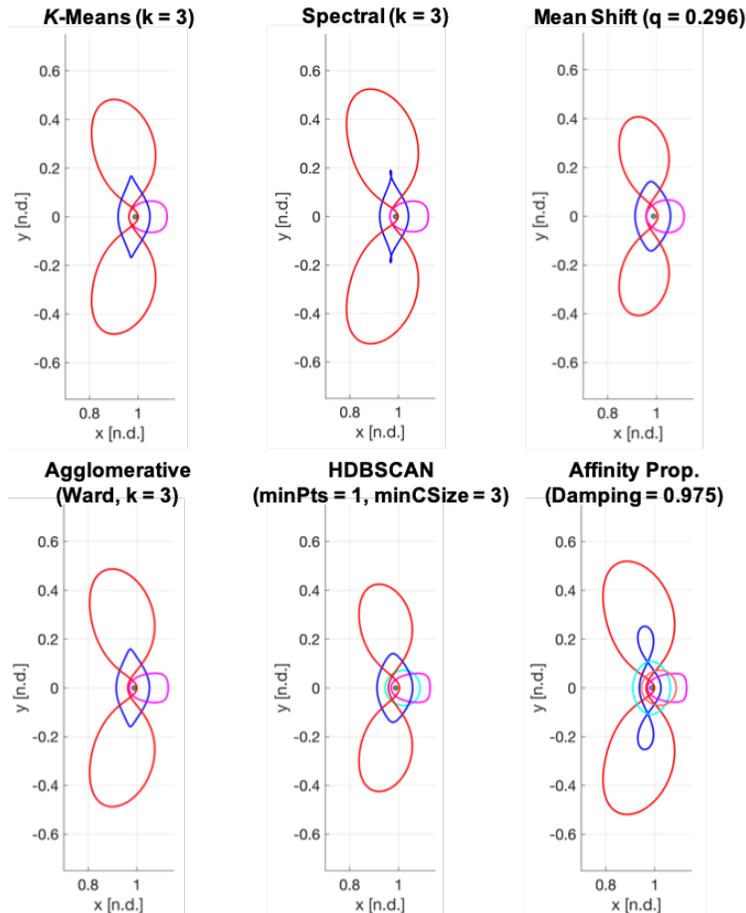
- *Affinity Propagation* Figure 11d) displays the number of clusters produced by applying affinity propagation to the DPO family for a range of damping factors from 0.88 to 0.98. Small variations in the damping factor impact the resulting number of clusters, producing an instability in the number of clusters. However, for damping factors around 0.975 there is little variation in the number of clusters; thus, 0.975 is selected from this stable region of the parameter search.

- *Agglomerative Clustering* Ward linkage is used in this analysis. From the dendrogram produced by the algorithm, an input of 3 clusters is selected as it appears to be the dominant number of clusters in the hierarchy before a significant increase in the number of clusters occurs as the inter-cluster distance decreases.

- *HDBSCAN* HDBSCAN is applied to the DPO family over an array of *minClusterSize* and *minPts* values. Similar to the agglomerative clustering analysis, the number of clusters produced by the algorithm for each pair of values is utilized in conjunction with the tree structures produced

by the algorithm to select appropriate values for the input parameters. From this qualitative analysis,  $minPts = 1$  and  $minClusterSize = 3$  were selected as appropriate inputs.

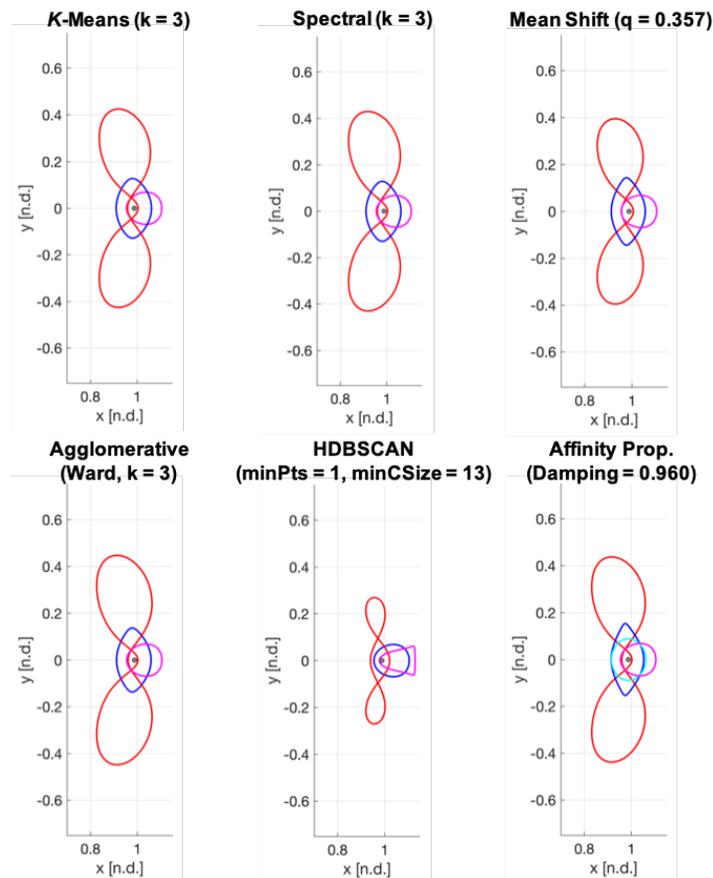
Using these input parameters, a set of motion primitives is constructed from the clustering results produced by each algorithm as displayed in Figure 12. For each algorithm, the selected input parameters are included in the caption and each motion primitive is plotted in a different color. All six algorithms produce similar motion primitives for the family of DPOs using the geometric feature space. The three primitives constructed using  $k$ -means and agglomerative clustering with Ward linkage are nearly identical. Furthermore, these solutions are similar to the primitives constructed using spectral clustering and mean shift. In fact, the motion primitives constructed by each algorithm are consistent but, due to their centroid-based definition, the primitives vary slightly because there are small variations in the size and shape of the clusters produced by each algorithm. HDBSCAN and affinity propagation produce four and five motion primitives, respectively, because both algorithms identify additional clusters at the boundaries between the three main clusters identified by the other algorithms. The boundaries between clusters are not clearly defined and there are no analytical expressions to determine these boundaries. Therefore, each set of motion primitives may be an acceptable summary of the finite geometries present in the family of DPOs.



**Figure 12.** Sets of motion primitives constructed from the results of each clustering algorithm when applied to the family of DPOs using a geometric feature space.

## Reduced Parametric Feature Space

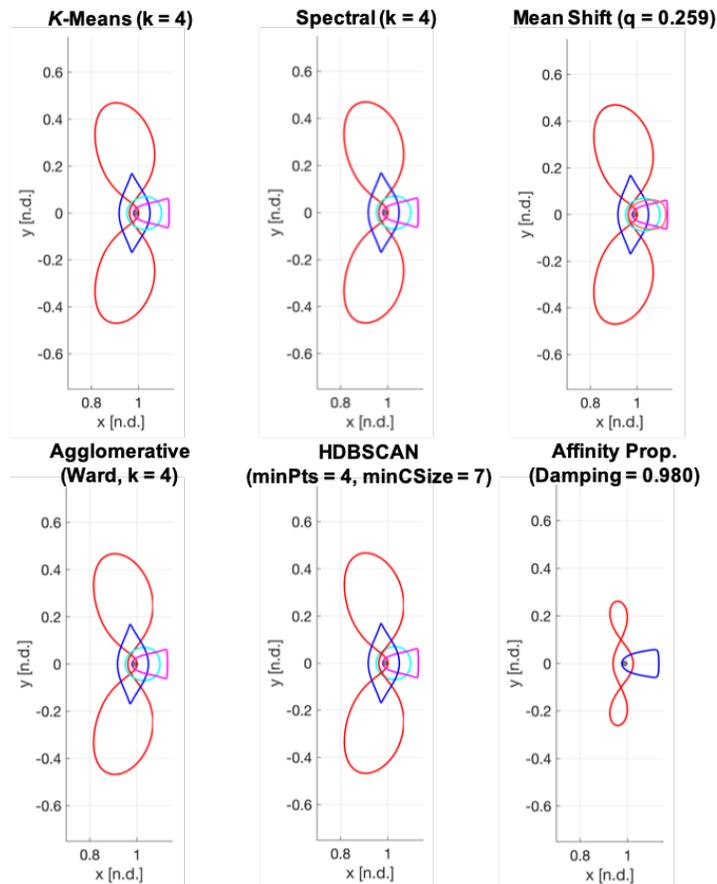
A large feature space may either increase the computational complexity of each clustering algorithm or diminish the importance of each feature; thus, a reduction in dimensionality is desirable. A reduced parametric feature space is defined for the DPO family using the feature vector defined in Eq. 8, resulting in a three-dimensional feature space reflecting the Jacobi constant, orbit period, and number of apsides relative to the Moon. The heuristic input parameter analysis presented earlier is used to select appropriate input parameters for each algorithm using this new feature vector. Each clustering algorithm produces similar sets of motion primitives as depicted in Figure 13. Affinity propagation constructs four motion primitives and presents a slightly different, yet feasible summarization of the DPO family due to the discovery of an additional cluster. However, the most significant difference in the motion primitives produced by each algorithm is in the HDBSCAN results, where a smaller DPO with multiple loops is captured as a motion primitive. This result is a reflection of HDBSCAN assigning all of the DPOs with multiple loops to the same cluster, impacting the motion primitive that is constructed due to the use of a centroid-based selection method. However, the primitives extracted from the HDBSCAN results still capture the distinctly different geometries of the orbits in this family. In addition, Figures 12 and 13 present similar sets of motion primitives. Using different parameterizations of the dataset, the clustering algorithms produce fundamentally different clusters but similar motion primitives are extracted.



**Figure 13.** Sets of motion primitives constructed from the results of each clustering algorithm when applied to the DPO family using a reduced parametric feature space.

## Geometric and Stability Feature Space

In addition to geometric features, the stability of an orbit is a valuable property in the trajectory design process. A geometric and stability feature space is defined for the DPO family using the feature vector defined in Eq. 9, resulting in a 22-dimensional feature space. Similar to the preceding cases, the heuristic input parameter analysis is leveraged to select appropriate inputs for each algorithm. The sets of motion primitives constructed using each clustering algorithm are displayed in Figure 14 with the input parameters summarized in the captions. These results generally suggest the existence of four clusters in the dataset.  $K$ -means, spectral clustering, agglomerative clustering, and HDBSCAN produce nearly identical motion primitives. Each algorithm successfully identifies the stability differences between the clusters corresponding to the red, cyan, and magenta primitives. A distinction is also made between the clusters corresponding to the blue and magenta primitives due to their different geometries despite their identical stability indices. The mean shift algorithm also produces similar motion primitives but recovers an additional primitive due to a small variation in stability within the oval-shaped cluster of DPOs. However, the results of affinity propagation seem to place more emphasis on the geometric features of each orbit. This algorithm found the similarities between orbit geometries to be more pronounced than the dissimilarities between stability classes throughout the dataset, producing two fundamental motion primitives. The results of these algorithms demonstrate different balances between the geometric and binary stability features.



**Figure 14.** Sets of motion primitives constructed from each clustering algorithm when applied to the DPO family using a geometric and stability feature space.

## CONCLUSION

Rapid trajectory design in the CR3BP relies on effectively leveraging the large variety of solutions that exist in one- or multi-dimensional families throughout the phase space of the system. While analysis and visualization of the solution space is complex, it may be simplified through the use of data analysis techniques. In this paper, motion primitives are constructed in the CR3BP via clustering algorithms to develop a simplified representation of the solutions across a family of periodic orbits. This analysis demonstrates the summarization of an entire family of DPOs in the Earth-Moon system via a small set of motion primitives constructed based on the geometry, stability, and characteristics of each orbit. Different parameterizations of the DPO family produce different natural clusterings. Therefore, the constructed set of motion primitives depends on the clustering algorithm and input parameter selection. Further analysis of motion primitive construction in the CR3BP is required to evaluate additional feature spaces on periodic orbit families and other dynamical structures in the CR3BP. In addition, an analysis of the computational performance and stability of each clustering algorithm is warranted. However, the results presented in this paper motivate the exploration of effective strategies for enabling rapid and robust trajectory design in multi-body systems via motion primitives.

## ACKNOWLEDGMENT

This work is being completed at the University of Colorado Boulder. It is supported by the Dean's Graduate Assistantship from the University of Colorado Boulder, the Graduate Assistantship in Areas of National Need (GAANN) Fellowship in Critical Aerospace Technologies from the United States Department of Education, and the NASA Space Technology Research Fellowship from the National Aeronautics and Space Administration.

## REFERENCES

- [1] R. J. Whitley, D. C. Davis, L. M. Burke, B. P. McCarthy, R. J. Power, M. L. McGuire, and K. C. Howell, "Earth-Moon Near Rectilinear Halo and Butterfly Orbits for Lunar Surface Exploration," *AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT, August 2018.
- [2] R. L. Restrepo, R. P. Russell, and M. W. Lo, "Europa Lander Trajectory Design Using Lissajous Staging Orbits," *AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT, August 2018.
- [3] N. Bosanac, C. M. Webster, K. C. Howell, and D. C. Folta, "Trajectory Design for the Wide Field Infrared Survey Telescope Mission," *Journal of Guidance, Control and Dynamics*, Vol. 42, No. 9, 2019.
- [4] D. Guzzetti, N. Bosanac, A. Haapala, K. C. Howell, and D. C. Folta, "Rapid Trajectory Design in the Earth-Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Orbits," *66th International Astronautical Congress*, Jerusalem, Israel, October 2015.
- [5] A. Cox, N. Bosanac, D. Guzzetti, K. C. Howell, D. C. Folta, and C. M. Webster, "An Interactive Trajectory Design Environment Leveraging Dynamical Structures in Multi-Body Regimes," *6th International Conference on Astrodynamics Tools and Techniques*, Darmstadt, Germany, March 2016.
- [6] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. London, UK: Academic Press, 1967.
- [7] A. A. Paranjape, K. C. Meier, X. Shi, S. J. Chung, and S. Hutchinson, "Motion Primitives and 3-D Path Planning for Fast Flight Through a Forest," *The International Journal of Robotics Research*, Vol. 34, No. 3, 2015, pp. 357–377.
- [8] A. Hadjighasem, D. Karrasch, H. Teramoto, and G. Haller, "Spectral-Clustering Approach to Lagrangian Vortex Detection," *Physical Review E*, Vol. 93, No. 6, 2016.
- [9] N. Nakhjiri and B. F. Villac, "Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design," *Celestial Mechanics and Dynamical Astronomy*, Vol. 123, No. 1, 2015, pp. 63–83.
- [10] B. F. Villac, R. L. Anderson, and A. J. Pini, "Organizing Ballistic Orbit Classes Around Small Bodies," *The Journal of the Astronautical Sciences*, Vol. 63, No. 3, 2016, pp. 175–205.

- [11] N. Bosanac, "A Data Mining Approach to Using Poincaré Maps in Multi-Body Trajectory Design Strategies," *29th AAS/AIAA Spaceflight Mechanics Meeting*, Ka'anapali, HI, January 2019.
- [12] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*. Waltham, MA: Morgan Kaufmann Publishers, 3 ed., 2012.
- [13] N. Bosanac, *Leveraging Natural Dynamical Structures to Explore Multi-Body Systems*. PhD dissertation, Purdue University, West Lafayette, IN, 2016.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.
- [15] U. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, Vol. 17, No. 4, 2007, pp. 395–416.
- [16] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems*, MIT Press, 2001, pp. 849–856.
- [17] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, 1995, pp. 790–799.
- [18] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, 2002, pp. 603–619.
- [19] Y. Ukrainitz and B. Sarel, "Mean Shift Theory and Applications," [www.wisdom.weizmann.ac.il/~vision/courses/2004\\_2/files/mean\\_shift/mean\\_shift.ppt](http://www.wisdom.weizmann.ac.il/~vision/courses/2004_2/files/mean_shift/mean_shift.ppt), 2004.
- [20] 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*, Springer Berlin Heidelberg, 2013, pp. 160–172.
- [21] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical Density Based Clustering," *The Journal of Open Source Software*, Vol. 2, Mar 2017.
- [22] B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, Vol. 315, No. 5814, 2007, pp. 972–976.
- [23] J. S. Parker, K. E. Davis, and G. H. Born, "Chaining Periodic Three-Body Orbits in the Earth-Moon System," *Acta Astronautica*, Vol. 67, 2010, pp. 623–638.