

IAC-25,C1,7,3,x96326

Using Motion Primitives to Design Thrust-Enabled Trajectories in the Earth-Moon System

Natasha Bosanac^{a*}

^a *Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO, USA 80303*

* Corresponding Author

Abstract

When a spacecraft operating in cislunar space possesses a continuous-thrust propulsion system, the trajectory design process becomes more complex. To address this challenge, this paper applies a motion primitive approach to continuous-thrust trajectory design. First, motion primitives are generated to summarize continuous-thrust arcs that approach or depart selected periodic orbits in the Earth-Moon circular restricted three-body problem. These thrust arcs use a constant thrust direction in the velocity-normal-conormal axes defined relative to the Moon, as well as a specified spacecraft with various initial masses. Then, a hierarchical motion primitive graph is used to summarize their potential for sequential composability in the phase space and mass. Each component of this graph is searched to generate multiple traversable primitive sequences. These primitive sequences are translated into initial guesses and corrected to produce continuous thrust-enabled trajectories. Distinct sequences produce geometrically distinct initial guesses with various propellant mass requirements and flight times. This approach is demonstrated in the foundational problem of designing planar transfers from an L_1 Lyapunov orbit to an L_2 Lyapunov orbit.

Keywords: motion primitive, continuous-thrust trajectories, trajectory design, multi-body system

1. Introduction

When a spacecraft operating in cislunar space possesses a continuous-thrust propulsion system, the trajectory design process becomes more complex. Traditionally, trajectory design within multi-body systems has relied on fundamental solutions, computed via dynamical systems theory, in low-fidelity models to support manual initial guess construction [1]. However, in the presence of an additional acceleration with an evolving direction and/or magnitude, motion is no longer governed by fundamental solutions. Accordingly, these existing approaches offer limited support for designing and predicting thrust-enabled trajectories.

In recent years, Smith and Bosanac have introduced a motion primitive approach to spacecraft trajectory design within multi-body gravitational systems [2,3]. This work has since been substantially extended by Miceli and Bosanac [4], as well as Gillespie, Miceli, and Bosanac [5]. Consistent with their use in robotics, motion primitives supply building blocks of motion that can be assembled to form complex paths [6-8]. When applied to spacecraft trajectory design, a set of arcs are sampled from trajectories with various predefined behaviors or parameters [2,4,5]. These arcs are clustered based on geometric similarity [4,5]. A single representative member of each cluster then serves as the motion primitive [2]; behavioral motion primitives also encode the associated behaviors and parameters [5]. A motion primitive graph is constructed to capture the sequential composability of pairs of these primitives [3,4]. This graph is then searched to generate unique primitive sequences that can supply geometrically distinct initial

guesses [3,4]. This approach has previously been used to design complex trajectories with impulsive maneuvers in the Earth-Moon [3] and Neptune-Triton systems [4].

This paper focuses on using this motion primitive approach to design low-thrust trajectories for a SmallSat in cislunar space. First, a library of motion primitives is generated to summarize continuous-thrust trajectories approaching or departing selected libration point orbits with various spacecraft masses and thrust directions in the Earth-Moon circular restricted three-body problem (CR3BP). Then, a motion primitive graph is constructed and searched to automatically generate geometrically distinct initial guesses for planar trajectories from an L_1 Lyapunov orbit to an L_2 Lyapunov orbit. This approach is intended to support exploring the associated trade spaces for spacecraft with low-thrust propulsion systems operating in cislunar space.

2. Background

2.1 Dynamical Model

The dynamical environment governing the motion of a spacecraft in cislunar space is approximated using the Earth-Moon CR3BP, with a continuous-thrust acceleration added. In this model, the Earth and Moon are modeled using constant masses with gravitational fields identical to point masses [9]. In addition, the spacecraft is assumed to possess a negligible mass of m in comparison. Finally, the Earth and Moon are assumed to follow circular orbits about their barycenter [9].

In this paper, the spacecraft is assumed to be a SmallSat with a continuous-thrust propulsion system. The initial wet mass of the spacecraft is between 170-180

kg with up to 40 kg of propellant [10]. The propulsion system is assumed to apply a constant thrust magnitude $T = 13mN$ and constant specific impulse $I_{sp} = 1390s$, consistent with a Busek BHT-200 engine [11].

The state of a spacecraft in the Earth-Moon CR3BP is typically expressed using nondimensional coordinates in a frame that rotates with the Earth and Moon [9]. Length, time, and mass quantities are normalized by the following characteristic quantities: $l^* = 384,400$ km, $t^* \approx 3.751903 \times 10^5 s$, and m^* . These values result in the distance between the Earth and Moon, the mean motion of the Earth-Moon system, and total system mass all possessing nondimensional values of unity [9]. The Earth-Moon rotating frame is defined using an origin at the barycenter of the Earth-Moon system and the following axes: $\hat{x}$ is directed from the center of the Earth to the center of the Moon, $\hat{z}$ is aligned with the orbital angular momentum vector of the Earth and Moon, and $\hat{y}$ completes the orthogonal right-handed triad [9].

The equations of motion governing a spacecraft with a continuous-thrust propulsion system are written in the rotating frame using nondimensional coordinates [9]. The state of the spacecraft is defined as $[x, y, z, \dot{x}, \dot{y}, \dot{z}, m]^T$. The dot notation indicates a time derivative with an observer fixed in the rotating frame. Using these definitions, the differential equations governing the spacecraft are written as [9,12]

$$\begin{aligned}\ddot{x} - 2\dot{y} &= \frac{\partial U^*}{\partial x} + \frac{T}{m}u_x \\ \ddot{y} + 2\dot{x} &= \frac{\partial U^*}{\partial y} + \frac{T}{m}u_y \\ \ddot{z} &= \frac{\partial U^*}{\partial z} + \frac{T}{m}u_z \\ \dot{m} &= -\frac{Tt^*}{I_{sp}g_0}\end{aligned}$$

where $\mu = 1.215058439470971 \times 10^{-2}$ is the mass ratio of the Earth-Moon system; u_x, u_y, u_z are the components of the thrust vector in the axes of the Earth-Moon rotating frame; $r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$; and g_0 is the gravitational acceleration on the surface of Earth. The final equation produces a quantity in kg per nondimensional time units. In the natural CR3BP, the Jacobi constant is defined as

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

and supplies a constant of motion.

2.2 Thrust Vector Definition

When the spacecraft propulsion system is activated, the thrust vector direction must be specified. In this paper, the velocity-normal-conormal (VNC) axes are used and defined in the inertial frame relative to the Moon. Consider the nondimensional position and velocity vectors of the spacecraft relative to the Moon in

the inertial frame, defined as $\bar{R}_{L,sc}$ and $\bar{V}_{L,sc}$. The VNC axes relative to the Moon are calculated as

$$\hat{V} = \frac{\bar{V}_{L,sc}}{V_{L,sc}}, \quad \hat{N} = \frac{\bar{R}_{L,sc} \times \bar{V}_{L,sc}}{\|\bar{R}_{L,sc} \times \bar{V}_{L,sc}\|}, \quad \hat{C} = \hat{V} \times \hat{N}$$

In the two-body problem, these axes supply an intuitive representation of the thrust vector based on strategically changing the energy or the orbit plane relative to the Moon. However, geometric intuition is limited in the CR3BP where multiple celestial bodies gravitationally interact with the spacecraft. An additional limitation is that a singularity exists when the position and velocity vectors of the spacecraft, relative to the Moon and in the inertial frame, are parallel. The thrust vector is defined in these axes as $\hat{u} = u_v \hat{V} + u_n \hat{N} + u_c \hat{C}$. Then, the thrust vector is transformed into the rotating frame at each epoch during numerical integration.

2.3 Curvature

Concepts from differential geometry are useful for describing and sampling curved paths in three-dimensional space [13,14]. Consider a path generated over a time interval $t \in [t_0, t_f]$ with position, velocity, and acceleration vectors denoted as $\bar{r} = [x, y, z]^T$, $\bar{v} = [\dot{x}, \dot{y}, \dot{z}]^T$, and $\bar{a} = [\ddot{x}, \ddot{y}, \ddot{z}]^T$. The path traverses a distance equal to the arclength, calculated as [13]

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

At any state along the curved path at time t , the scalar, unsigned curvature captures the deviation from a straight line. This quantity is calculated as [14]

$$\kappa(t) = \frac{\|\bar{v} \times \bar{a}\|}{\|\bar{v}^3\|}$$

with a singularity when the speed is equal to zero. Maxima in this curvature occur when $\dot{\kappa}(t) = 0$, indicating that the shape is changing most rapidly [14].

2.3 Density-Based Clustering

To discover groupings of geometrically similar arcs, density-based clustering is employed. This unsupervised method identifies groups of data points that exist in sufficiently dense regions within a feature vector space [15]. This paper uses two clustering methods: Density-Based Clustering of Spatial Applications with Noise (DBSCAN) [16] and Hierarchical Density-Based Clustering of Spatial Applications with Noise (HDBSCAN) [17].

DBSCAN constructs clusters from data points that possess at least a specified number of members within a fixed radius. First, for each member of a dataset, its m_{pts} -neighborhood is defined to possess a radius equal to the distance to its m_{pts} neighbor in the feature vector space. Then, three types of points are defined [16]:

1. A core point possesses an m_{pts} -neighborhood with a radius that is less than or equal to ϵ .

2. A border point lies within a distance of ϵ of a core point but does not possess at least m_{pts} neighbors in its own ϵ -sized neighborhood.
3. A noise point does not lie within a distance of ϵ of any core points.

Then, each cluster is formed by core points that lie within a distance of ϵ from other core points, along with their border points [16]. Noise points are not assigned to any clusters. A modification of this algorithm, spatio-temporal DBSCAN (ST-DBSCAN) [18], updates the condition of a core point to possess at least m_{pts} neighbors within a radius of ϵ_1 in one feature vector space and ϵ_2 in a second feature vector space. DBSCAN is implemented in this paper using the built-in *dbscan* function in MATLAB [19].

HDBSCAN selects clusters from a hierarchy [17], eliminating the dependence on a single, constant value of ϵ . To achieve this goal, the distance between two points is transformed to a mutual reachability distance to capture density information [17]. First, the core distance of the i th member of a dataset, $d_{core}(\bar{f}_i)$, is the distance to its m_{pts} th nearest neighbor. The mutual reachability distance is then defined between members i and j as

$$d_{mrd}(\bar{f}_i, \bar{f}_j) = \max(d_{core}(\bar{f}_i), d_{core}(\bar{f}_j), d(\bar{f}_i, \bar{f}_j))$$

where $d(\bar{f}_i, \bar{f}_j)$ is the Euclidean distance between the two points in the specified feature vector space. A cluster hierarchy is then constructed to summarize all possible groupings as the threshold on this mutual reachability distance is varied, analogous to varying ϵ in DBSCAN. The selected clusters are most persistent across the hierarchy with at least $m_{minclust}$ members [17]. A threshold ϵ_{merge} can also be defined to limit the distance between members in distinct clusters [20]. Through this clustering approach, members are either assigned to a unique cluster or designated as noise. HDBSCAN is accessed using the Python-based *hdbscan* library [21].

3. Technical Approach

This section supplies a brief overview of the motion primitive approach to designing continuous-thrust transfers. The motion primitive library is constructed by following an approach recently presented by Gillespie, Miceli, and Bosanac [5]. Then, the remaining steps of this framework follow the general procedure developed by Miceli and Bosanac [4]. Recent modifications to reduce computational complexity in the graph construction step were outlined by Bosanac [22].

3.1 Step 1: Generate Motion Primitive Library

The motion primitive library is constructed to summarize the prominent geometries exhibited across planar trajectories that approach or depart specified families of periodic orbits. In this paper, the selected periodic orbits are 31 L_1 Lyapunov orbits and 12 members of the L_2 Lyapunov orbit families with a Jacobi

constant greater than 3.0 and less than 3.2. Trajectories that approach or depart these periodic orbits are generated in the Earth-Moon CR3BP for a duration equal to the perturbation doubling time [23] plus three months [5]. However, propagation is terminated early if 1) the trajectories impact the Earth or Moon; 2) the trajectories exceed a distance of 1 from the Moon [5]; or 3) the position and velocity vectors relative to the Moon are within 0.1 degrees of being parallel or anti-parallel, corresponding to a singularity in the definition of the VNC axes. Trajectories that approach or depart any periodic orbit with continuous thrust are generated by assuming 1) a constant thrust vector in the VNC axes defined relative to the Moon and 2) a specific initial mass at the arrival or departure location along the periodic orbit [5]. The thrust vector is varied in 45 degree increments within the VC-plane and the initial mass is varied in 1 kg increments between 180 kg and 170 kg.

The trajectories that approach or depart the selected periodic orbits are sampled to produce smaller arcs, following the approach presented by Gillespie, Miceli, and Bosanac [5]. These arcs are defined using curvature-based windows to ensure consistency as the trajectories visit various regions of the system. Each window that defines an arc begins at either a curvature maximum or the initial state along a trajectory. The window is then defined to encompass three more curvature maxima and either a fourth curvature maximum or an early termination state. If termination occurs before three more curvature maxima, the window is shortened to produce a smaller arc. When sampled from a long trajectory, one arc begins where the previous arc ends.

Each continuous arc is then sampled using a discrete sequence of states that are distributed geometrically [5]. First, the initial and final states along with the intermediate curvature maxima are retained. Then, two additional samples are added between subsequent samples and distributed equally in the arclength [5]. Accordingly, each arc is sampled using up to 13 states.

The state vectors at the discretely sampled states along an arc are used to construct two finite-dimensional feature vectors that captures its geometry [5,24]. A position-based feature vector is defined using the position vectors in the Earth-Moon rotating frame at these samples as [5,24]

$$\bar{f}_p = [\bar{r}_1, \bar{r}_2, \dots, \bar{r}_{N_f-1}, \bar{r}_{N_f}]$$

In addition, a shape-based feature vector is defined using the velocity unit vectors in the Earth-Moon rotating frame at these samples as [5,24]

$$\bar{f}_s = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_{N_f-1}, \hat{v}_{N_f}]$$

Each feature vector is $3N$ -dimensional for an arc sampled using $N \leq 13$ states.

Clustering is used to construct a coarse, initial grouping of these arcs by their shape [24]. At this stage, the arcs sampled from trajectories that either approach or

depart a single periodic orbit with any thrust vector and any initial mass are identified [5]. This set is further subdivided to produce groups of arcs that are sampled by the same number of states. Partitions of up to 10,000 arcs are constructed from each group. Then, the shape-based feature vectors are input to HDBSCAN for initial, coarse clustering. In this paper, the governing parameters of this algorithm are selected to prioritize discovering localized geometric variations between arcs: $m_{pts} = 19$, $m_{minclust} = 20$, and $\epsilon_{merge} = 2 \sin(10^\circ)$. In addition, the Euclidean distance is used to compare two feature vectors for computational efficiency. This step produces a set of coarse groups, labeled $\mathcal{C}^i$, for the i th set of arcs and noise points, labeled $\mathcal{N}^i$. However, because this step uses only the shape of the entire trajectories and the Euclidean distance between two sequences of vectors does not capture the time at which any differences occur, these coarse groups are further refined.

The cluster refinement process is used to generate clusters of trajectories that consistently follow a similar path in each other's neighborhoods for their entire duration. This procedure, developed by Bosanac [24], is modelled after convoy detection schemes from trajectory clustering and uses both feature vector spaces, similar to ST-DBSCAN [18]. Specifically, consider the coarse shape-based group $\mathcal{C}_j^i$. The k th state sampled along all q trajectories in this group is described by two three-dimensional feature vectors: $\bar{f}_p$ and $\bar{f}_s$. Then, DBSCAN is used to cluster the $q \bar{f}_p$ vectors to produce a single clustering result in the position-based feature vector space as well as record the feature vectors that lie in each other's neighborhoods. This step is repeated in $\bar{f}_s$ to produce a second clustering result for state k . This process is also repeated across all N states along the arc to produce $2N$ clustering results. If any two trajectories are consistently clustered together and lie in each other's neighborhoods in all $2N$ clustering results, the trajectories are assigned to a candidate cluster. Then, if at least $m_{minclust}$ trajectories are grouped together, the candidate cluster is retained as a refined cluster [24]. This process produces a set of refined clusters $\mathcal{R}^i$ for the i th set of arcs and updated noise points $\mathcal{N}^i$. Noise points are discarded at this step.

When using DBSCAN in the cluster refinement process, the governing parameters must be selected. First, $m_{pts} = 19$ to ensure consistency with the coarse clustering step when defining the neighborhood membership that governs the assessment of density. In addition, the neighborhood radius ϵ is selected heuristically based on the data in the coarse cluster [24]. This value is challenging to select with traditional methods, e.g., the elbow method, to balance avoiding excessive differentiation with excessive overgrouping. In this paper, the heuristic is defined as follows: $\epsilon =$

$\max(\min(e_1 m_{pts}, e_{mpts}), \epsilon_{threshold})$. In the first term, e_1 and e_{mpts} are the $m_{minclust}$ -largest value of the distance to the first and m_{pts} th nearest neighbor, respectively. Accordingly, the first term estimates a suitable radius for the m_{pts} -neighborhood based on the actual data while avoiding biasing from outliers [24]. Then, $\epsilon_{threshold}$ is defined as $2 \sin(5^\circ)$ and 1×10^{-2} in the shape- and position-based feature vector spaces to encode a minimum tolerance on the separation between two trajectories.

Each refined cluster is used to generate a motion primitive and its region of existence [2,4]. The motion primitive is a single trajectory that summarizes the geometry of the cluster's members. This trajectory is selected as the cluster medoid calculated using $\bar{f}_p$ evaluated over the entire trajectory [4]. The region of existence summarizes the region of the phase space spanned by arcs with a similar geometry. This region of existence is approximated by up to 20 trajectories sampled from the cluster and roughly evenly distributed in the position-based feature vector space [4].

To produce a condensed summary, the clusters from each set of arcs is then aggregated across an entire dataset. Aggregation is performed across arcs approaching or departing neighboring periodic orbits along the same family with various thrust vectors and initial masses. This procedure is performed using the approach developed by Bosanac [24]. First, the motion primitives of clusters from distinct sets are compared coarsely using their position and shape-based feature vectors. The nearest neighboring clusters to the i th cluster in each feature vector space from sets of arcs approaching the same or neighboring periodic orbits are then considered as two candidates for aggregation. All members of each neighbor candidate and the original cluster are then input to the cluster refinement process. If any trajectories from two distinct clusters are grouped together, the clusters are merged. These pairwise merging decisions are then used to identify groups of local clusters that are connected to form a single global cluster. This global cluster consists of arcs that are geometrically similar but may approach or depart the same or neighboring orbits, may be generated with different control histories, or may use a distinct initial wet mass. The motion primitive and region of existence of each global cluster is recomputed.

This approach is used to summarize continuous-thrust, planar arcs that depart an L_1 Lyapunov orbit or approach an L_2 Lyapunov orbit. Each of the members of the L_1 Lyapunov and L_2 Lyapunov orbit families are discretized with at least 200 states or to produce a separation of no more than 10^{-2} in the configuration space. At each of these initial or final states, the spacecraft mass is varied within the range [170, 180] kg in increments of 1 kg. The thrust direction is also varied

in increments of 45 degrees. Once the thrust direction, mass, and initial state are specified, the trajectories are propagated forward or backward in time, discretized, and the arcs clustered.

This approach produces a library of 2,335 primitives summarizing motion departing an L_1 Lyapunov orbit and 1,539 primitives summarizing motion approaching an L_2 Lyapunov orbit. Examples of these primitives are displayed in Figure 1 for the arcs departing an L_1 Lyapunov orbit and Figure 2 for the arcs approaching an L_2 Lyapunov orbit. Analysis of Figures 1 and 2 reveals that, even in the presence of continuous thrust, the motion primitives successfully summarize arcs of similar geometry while separating arcs of distinct geometry. Sometimes this separation occurs with obvious differences in the shape or region of the configuration space that is traversed. In other cases, distinct clusters can be formed by similarly shaped and nearby groups of trajectories because they form localized dense regions with a sufficient separation between them. This separation is due to the small values selected for the governing parameters as well as the static sampling of a periodic orbit to define initial states and other parameters. Interestingly, the majority of individual clusters tend to include trajectories with a wide range of initial masses. This observation is consistent with the expectation that a SmallSat, applying a small continuous-thrust acceleration, may follow geometrically similar paths with only small differences in the initial mass.

3.2 Step 2: Identify Sequentially Composable Primitives

To support constructing complex paths from sequences of motion primitives, the sequential composability of two primitives must be defined. Consistent with prior work by Miceli and Bosanac [4], two primitives are considered sequentially composable if the primitives pass sufficiently close in the configuration space with a similar direction of motion. The sequential composability is then quantified using a cost function that reflects the change in velocity at each overlapping region.

Each region of existence is discretized into multiple segments to support identifying multiple overlapping locations [4,25]. These segments are defined via the geometrically distributed samples generated in Step 1 [4,25]. The i th segment along a single arc is defined between the i th and $(i+1)$ th sampled state. For an arc that completes three additional curvature maxima between the initial and final state and is, therefore, sampled with 13 states, 12 segments are formed. As a spacecraft traverses the trajectory forward in time, the segment number increases from 1 to 12. Repeating this approach for all representative trajectories that span the region of existence of a primitive, the associated volume of the phase space is discretized into 12 segments.

Each segment along a region of existence is coarsely approximated using a collection of states and circular

neighborhoods in the configuration space [4]. Specifically, the i th segments of all arcs in a region of existence are sampled using an additional four states that are equally distributed in arclength. Then, similar to the approach presented by Miceli and Bosanac [4], circular neighborhoods are defined in the configuration space around each sampled state with radii equal to the maximum of 1) the distance to their 2nd nearest neighbor, and 2) a threshold of 5×10^{-3} [22].

The coarse approximation of each region of existence is used to determine whether two primitives are sequentially composable. This definition is slightly adapted from the approach used by Miceli and Bosanac [4] to support thrust-enabled arcs. For continuous thrust trajectories, it is important to substantially limit any velocity changes between arcs along composable primitives. In this paper, primitives 1 and 2 are sequentially composable at their i th and j th segments if they satisfy the following conditions:

- 1) Any circular neighborhoods of the two segments overlap in the configuration space.
- 2) The velocity vectors of two overlapping neighborhoods possess an angular separation of less than 45 degrees, and a difference with a magnitude less than a specified tolerance.
- 3) The change in the Jacobi constant is below a specified threshold.

Note the difference in the spacecraft mass is not considered due to the limited sensitivity of most geometries to the initial spacecraft mass. Nevertheless, two primitives may be sequentially composable at multiple unique combinations of segments.

The cost of sequentially composing two primitives is quantified using the minimum difference in the velocity vectors at any composable states within the overlapping segments [22]. This cost to transition from segment i along primitive 1 to segment j along primitive 2 is defined as $w_{1,i,2,j}$ and calculated as

$$w_{1,i,2,j} = \min \left(\left| \bar{v}_{1,i} - \bar{v}_{2,j} \right| \right)$$

In this expression, the notation $\bar{v}_{1,i}$ indicates the velocity vector at the i th sample within segment i along primitive 1.

3.3 Step 3: Construct Motion Primitive Graph

A hierarchical motion primitive graph is constructed to summarize the sequential composability of each motion primitive and, therefore, construct traversable sequences of primitives. Due to the large number of motion primitives in the library, two graphs are created to support computational feasibility. First, a high-level graph is constructed to summarize the composability of primitives. This graph is modeled after the first implementation by Smith and Bosanac [3]. Then, a low-level graph is constructed to summarize the traversability of a primitive sequence. This graph is modeled after the

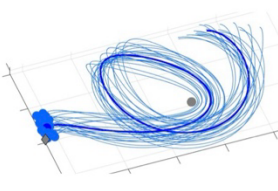
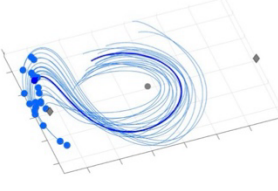
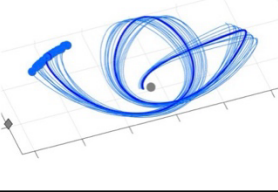
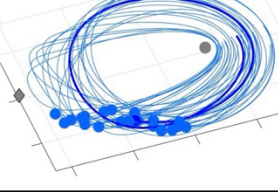
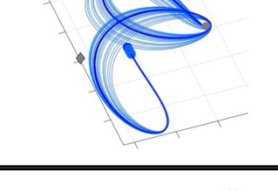
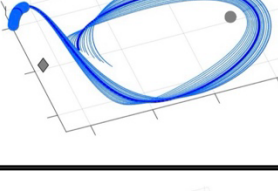
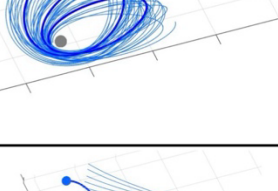
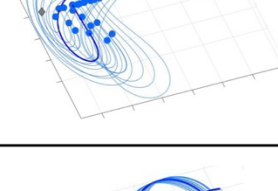
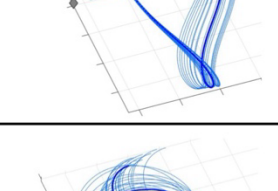
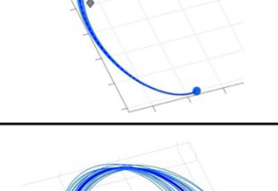
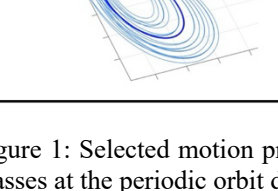
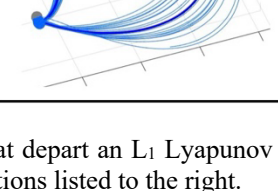
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$		m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}, -\hat{C},$ $-\sqrt{2}(\hat{V} + \hat{C})$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$		m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}$		m_0 $\in [173,177]kg$ $\hat{u} = -\hat{V}$
	m_0 $\in [170,179]kg$ $\hat{u} = -\hat{V},$ $-\sqrt{2}(\hat{V} + \hat{C})$		m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}$
	m_0 $\in [170,178]kg$ $\hat{u} = +\hat{V}$		m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}$		m_0 $\in [170,180]kg$ $\hat{u} = -\hat{V}$

Figure 1: Selected motion primitives summarizing arcs that depart an L₁ Lyapunov orbit with the associated initial masses at the periodic orbit departure state and thrust directions listed to the right.

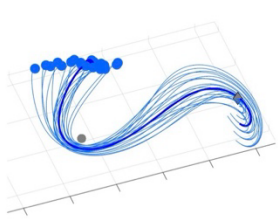
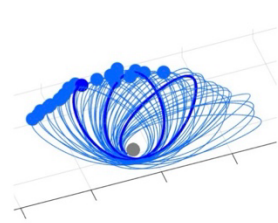
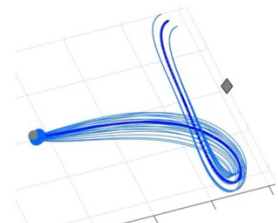
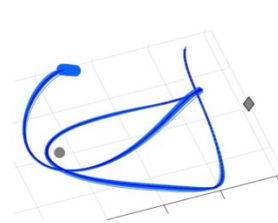
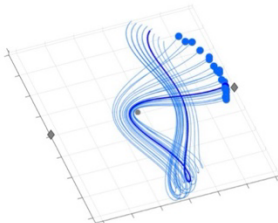
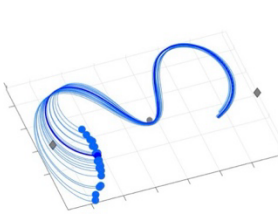
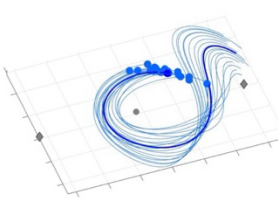
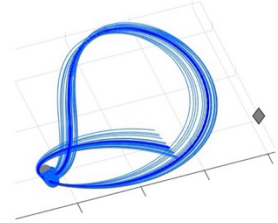
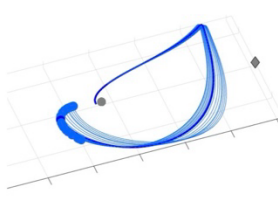
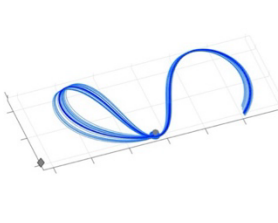
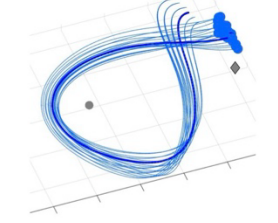
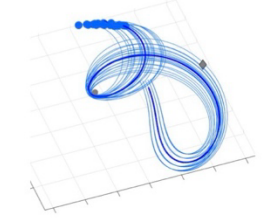
	m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\sqrt{2}(\hat{V} + \hat{C})$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$
	m_0 $\in [170,180]kg$ $\hat{u} = -\hat{C}$		m_0 $\in [170,180]kg$ $\hat{u} = +\hat{V}$

Figure 2: Selected motion primitives summarizing arcs that approach an L₂ Lyapunov orbit with the associated initial masses at the periodic orbit arrival state and thrust directions listed to the right.

more recent implementation by Miceli and Bosanac [4]. These two graph formulations were combined in a two-step process by Bosanac [22].

The high-level graph is used to generate candidate primitive sequences that are composable, without information about whether they are traversable, i.e., can a spacecraft travel forward in time along part of primitive 1, then part of primitive 2, followed by part of primitive 3 [4]. Accordingly, each node of this high-level graph represents a single primitive and its region of existence [3]. Bidirectional edges connect the nodes of primitives that are sequentially composable. The weight of an edge that connects two primitives is the minimum value of w along any combination of overlapping segments. Searching this high-level graph produces a candidate primitive sequence that requires further analysis [22].

The low-level graph is constructed to assess the traversability of a candidate sequence of primitives. Specifically, each node represents a segment of a primitive that appears in a candidate sequence. As formulated by Miceli and Bosanac [4], unidirectional, zero weight edges are added from the i th segment to the $(i+1)$ th segment along the same primitive. Then, an edge is added from a node correspond to a segment along the j th primitive to a node of a segment along the $(j+1)$ th primitive if those two segments are sequentially composable. In that case, the edge weight is defined as the value of w for the associated primitives and segments. If a solution exists, the result is a sequence of segments of primitives [4,22]. If a solution does not exist, that means that the candidate primitive sequence cannot be traversed in forward time.

3.4 Step 4: Search Motion Primitive Graph

To generate traversable primitive sequences, the hierarchical motion primitive graph is searched [22]. First, the high-level graph is searched using Dijkstra's algorithm to produce a candidate primitive sequence that minimizes the cumulative edge weight [22,25]. Then, additional candidate primitive sequences that produce suboptimal cumulative edge weights are generated using Yen's algorithm, a k -best paths search algorithm, similar to the approach presented by Miceli and Bosanac and used by Bruchko and Bosanac [4,22,25,26]. Each candidate primitive sequence is used to construct a low-level primitive graph. The low-level graph is searched using Dijkstra's algorithm to generate a single path that minimizes the cumulative edge weight [22]. The result is multiple sequences of segments of motion primitives that are traversable. Because the motion primitives are generated to be geometrically distinct, these sequences can produce geometrically distinct initial guesses.

Dijkstra's algorithm was developed by Edsger Dijkstra and produces a single sequence of nodes and edges through the graph that connects a start node to an end node [27]. First, a priority queue is initialized using

the start node with a zero cost. The lowest edge weight path in the priority queue is accessed and removed from the queue, with the last node in this incomplete path supplying the current node n_i to explore. The neighbors of n_i that have not yet been visited are identified using the edges of the graph. The cost of transitioning from n_i to a new node n_{i+1} is defined as $g(n_i, n_{i+1})$ where g is simply calculated from the edge weight. The cost to reach this new node from the start node, by following the path selected from the priority queue, is calculated as $g(n_s, n_{i+1}) = g(n_s, n_i) + g(n_i, n_{i+1})$. This new path and its cost are then added to the priority queue. This process is repeated until either the end node is reached, to produce a solution, or the priority queue is empty, producing no solution.

Yen's algorithm was developed by Jin Yen to generate the next $k-1$ best paths through a graph [28]. After the first path has been generated using Dijkstra's algorithm, the edge between subsequent nodes n_i and n_{i+1} in the path are removed from the full graph. Node n_i is labelled the spur node whereas the segment of the path from the start node to node n_i is labeled the root path. This subgraph is searched using Dijkstra's algorithm to generate an optimal path from the spur node to the end node, labelled the spur path. This spur path is concatenated to the root path to produce a new path connecting the start and end nodes, with its cost calculated using the cumulative edge weights. Repeating this process for all α edges in the original path of length $\alpha + 1$ produces α subgraphs and up to α new solutions. The solution with the lowest cost is then added to a list as the next best path. This process is repeated until either $k-1$ best paths are generated, or no additional solutions are identified. However, after the first iteration, the subgraph is modified to remove all edges from node n_i that appear in any of the prior best paths with a shared root path.

3.5 Step 5: Generate Initial Guess

A traversable primitive sequence is used to generate an initial guess for a trajectory, if one exists. This procedure, developed by Miceli and Bosanac [4], is constructed as a localized graph using the representative arcs from the region of existence of each primitive in the sequence. If a solution is generated by searching this localized graph with Dijkstra's algorithm, an initial guess is formed as the sequence of segments along the representative arcs that globally minimize a function of the state discontinuity [4].

The localized graph is constructed to capture the sequential composability of the representative arcs from each region of existence in a sequence of traversable primitive segments [4]. Each node corresponds to one segment of one arc sampled across the region of existence of a primitive. Unidirectional, zero weight edges are added between nodes that lie along the same arc. No edges are added between distinct arcs sampled

from the same region of existence, to ensure that the associated portion of the initial guess retains the geometry of the primitive. Unidirectional, weighted edges are added between composable segments of arcs from two sequential primitives. These edges are weighted by the velocity discontinuity between states along composable segments.

365 Step 6: Correct Initial Guess

A multiple-shooting scheme is implemented using a free variable and constraint vector formulation to correct each initial guess to produce a continuous trajectory. In this paper, the continuous-thrust trajectory is discretized into n arcs. Each arc is then described by the initial state, initial mass, constant thrust, and integration time. The arcs are constrained to enforce state and mass continuity as well as nonzero integration times along arcs generated with the specified thrust magnitude. Natural arcs are also added to perform up to two revolutions near L_1 and L_2 , resembling Lyapunov orbits. However, the boundary conditions are not constrained. The variables describing the trajectory are updated via Newton's method until they

produce a constraint vector with a norm that is less than 10^{-10} . The result is a continuous trajectory.

4. Results

The presented technical approach is applied to a continuous-thrust trajectory design problem in the Earth-Moon CR3BP: generating initial guesses for planar trajectories that depart an L_1 Lyapunov orbit and approach an L_2 Lyapunov orbit at distinct values of the Jacobi constant. To generate these initial guesses, the motion primitive graph is searched using selected primitives in the library as the boundary conditions. Due to the limited influence of the spacecraft mass for a SmallSat on the trajectory geometry, the mass constraint is not applied to the sequential composability assessment. Accordingly, the spacecraft mass for the initial guess is assumed to vary linearly in time with no discontinuities between arcs from distinct primitives.

Figure 3 displays an initial guess between selected primitives, via its a) discontinuous path in the Earth-Moon rotating frame, along with b) the corrected trajectory and c) the associated control history via the

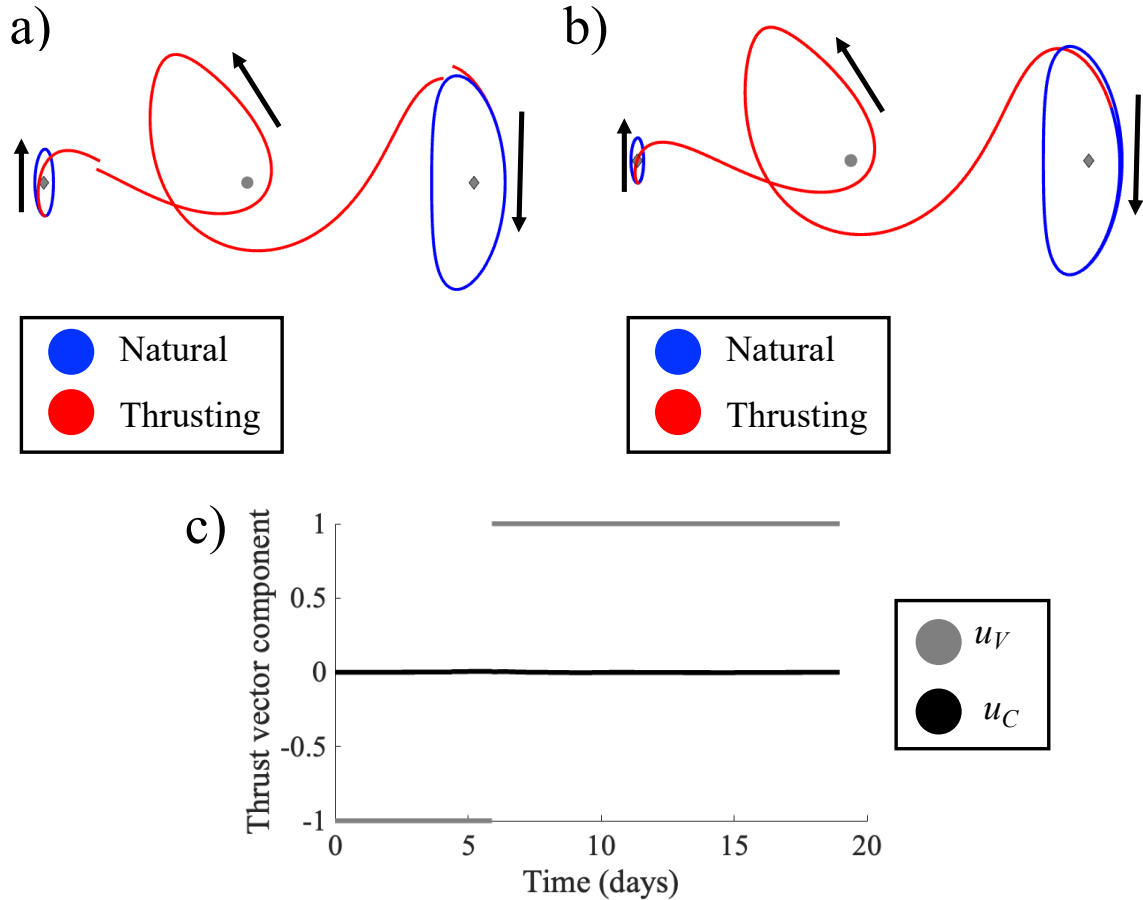


Figure 3: Trajectory departing an L_1 Lyapunov orbit and approaching a higher-energy L_2 Lyapunov orbit with continuous thrust: a) initial guess, b) corrected trajectory, c) thrust vector direction for corrected trajectory.

component of the thrust unit vector in the velocity and conormal directions, relative to the Moon. In Figures 3 a) and b), natural arcs are plotted in blue whereas thrust-enabled arcs are colored red. In this example, the corrected trajectory geometrically resembles the initial guess; note, however, that the size of the revolution around L_1 changed through corrections, consistent with the absence of required boundary conditions. In Figure 3 c), the velocity component of the thrust vector is plotted in gray whereas the conormal component is plotted in blue. Initially, the thrust vector is directed along the anti-velocity direction to decrease the spacecraft energy relative to the Moon. Then, the thrust vector is directed in the velocity direction, to increase the spacecraft energy relative to the Moon. This trajectory requires 1.56 kg of propellant, for an initial wet mass of nearly 180 kg, and a transfer time of 18.97 days.

Figure 4 displays two additional corrected transfers between selected primitives with distinct geometries. In Figure 4a), the transfer completes one revolution around

the Moon but with a distinct geometry compared with Figure 3. This trajectory requires 1.39 kg of propellant over a transfer time of 16.83 days. In Figure 4 b), the path completes six revolutions around the Moon, gradually varying the orientation of the line of apsides as well as the perilune and apolune distances. This trajectory requires 4.36 kg of propellant with a transfer time of 52.9 days. Across these examples, the transfers possess geometrically distinct paths, consistent with their construction from distinct motion primitive sequences.

6. Conclusions

This paper used a motion primitive approach to generate initial guesses for planar, continuous-thrust trajectories. First, motion primitives were generated using density-based clustering to summarize geometrically similar arcs that approach or depart libration point orbits with various thrust directions and initial masses. The sequential composability of a pair of primitives is then calculated by assessing their proximity

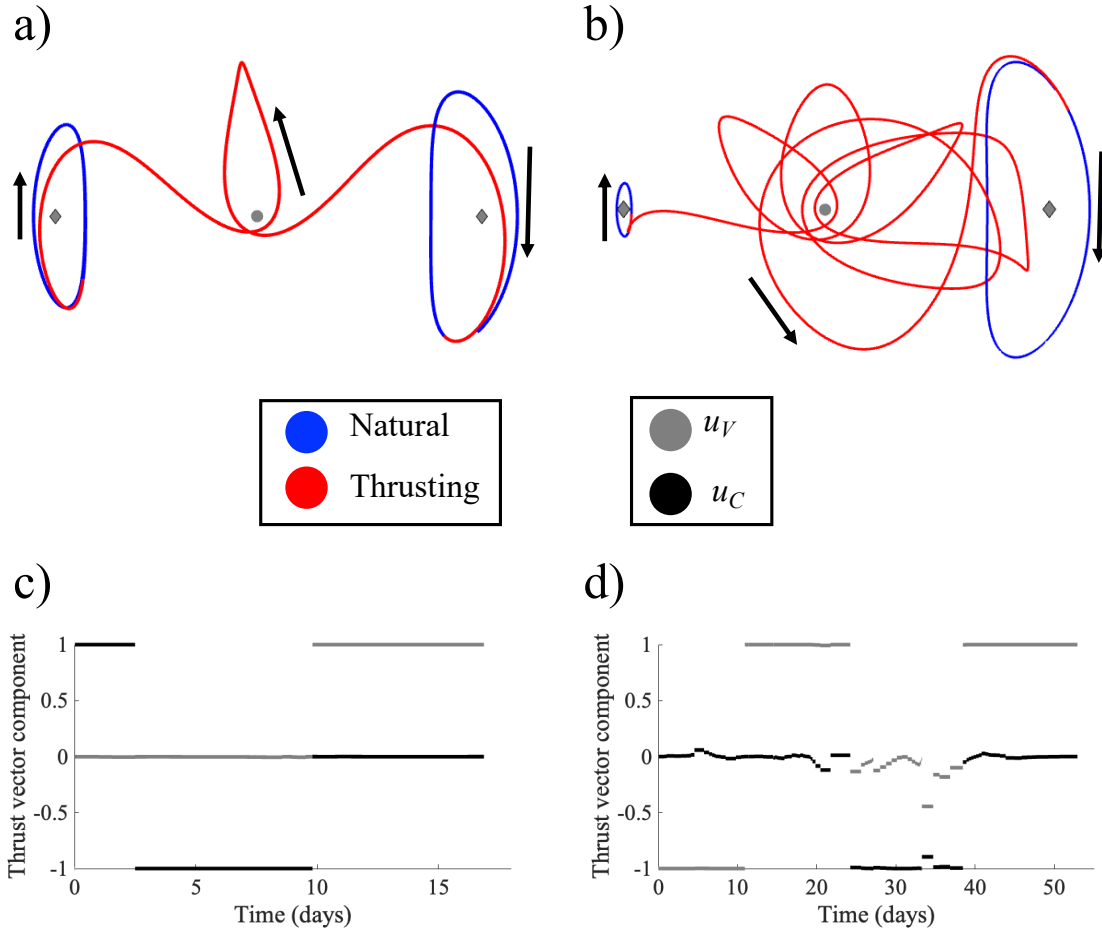


Figure 4: Selected initial guesses departing an L_1 Lyapunov orbit and approaching a higher-energy L_2 Lyapunov orbit with continuous thrust.

in the phase space. A hierarchical motion primitive graph is then constructed to discretely summarize the continuous trajectory design problem. Searching each of the two levels of this hierarchical motion primitive graph can produce a sequence of segments along motion primitives that connect an initial and target orbit. These primitive sequences are used to generate initial guesses for both the trajectory and their control history. Multiple-shooting is then used to correct each initial guess to produce a continuous path. This approach is used to design multiple, geometrically distinct and continuous-thrust transfers from natural motion near an L_1 Lyapunov orbit to natural motion around an L_2 Lyapunov orbit in the Earth-Moon CR3BP.

Acknowledgements

Part of this paper 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] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, *Dynamical Systems, the Three Body Problem and Space Mission Design*. New York: Marsden Books, 2011.
- [2] T.R. Smith and N. Bosanac, “Constructing Motion Primitive Sets to Summarize Periodic Orbit Families and Hyperbolic Invariant Manifolds in a Multi-Body Systems,” *Celestial Mechanics and Dynamical Astronomy*, vol. 134, no. 7, 2022.
- [3] T.R. Smith and N. Bosanac, “Motion Primitive Approach to Spacecraft Trajectory Design in a Multi-Body System,” *The Journal of the Astronautical Sciences*, vol. 70, no. 34, 2023.
- [4] G. Miceli and N. Bosanac, *Generating Trajectories for Neptunian System Exploration Using Motion Primitives*, Under review, 2025.
- [5] C. Gillespie, G. Miceli, and N. Bosanac, “Summarizing Natural and Controlled Motion in Cislunar Space with Behavioral Motion Primitives,” *AAS/AIAA Space Flight Mechanics Meeting*, Kaua’i, HI, Jan. 2025.
- [6] A. Wolek and C. Woolsey, “Model-Based Path Planning,” in *Sensing and Control for Autonomous Vehicles: Application to Land, Water and Air Vehicles*. Berlin: Springer, 2017.
- [7] E. Frazzoli, “Robust Hybrid Control for Autonomous Vehicle Motion Planning,” PhD Dissertation, Massachusetts Institute of Technology, Cambridge (2001)
- [8] D.J. Grymin, C.B. Neas, M. Farhood, “A Hierarchical Approach for Primitive-Based Motion Planning and Control of Autonomous Vehicles,” *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 214–228, 2014.
- [9] V. Szebehely, “Theory of Orbits: The Restricted Problem of Three Bodies,” London: Academic Press, 1967.
- [10] I. Elliott, C. Sullivan, N. Bosanac, J. Stuart, and F. Alibay, “Designing Low-Thrust Trajectories for a SmallSat Mission to Sun-Earth L_5 ,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 10, 2020.
- [11] Busek, BHT-200 Busek Hall Effect Thruster, Last accessed: June 28, 2024. [Online]. Available: <https://www.busek.com/bht200>.
- [12] B. Conway, “Spacecraft Trajectory Optimization,” New York: 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,” E-book, 2009.
- [15] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” 2nd ed. New York, NY: Proquest EBook Central: Elsevier Science and Technology, 2006, ch. 7
- [16] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996.
- [17] R. Campello, D. Moulavi, and J. Sander, “Density-Based Clustering Based on Hierarchical Density Estimates,” in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., Heidelberg: Springer, Berlin, 2013.
- [18] D. Birant and A. Kut, “ST-DBSCAN: An Algorithm for Clustering Spatial-Temporal Data,” *Data & Knowledge Engineering*, vol. 60, pp. 208–221, 2007.
- [19] The MathWorks Inc., MATLAB version: 9.14 (R2023a), Natick, Massachusetts, United States, 2023.
- [20] C. Malzer and M. Baum, “A Hybrid Approach To Hierarchical Density-Based Cluster Selection,” in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2020, pp. 223–228.
- [21] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical Density Based Clustering,” *Journal of Open Source Software*, vol. 2, no. 11, 2017.
- [22] N. Bosanac, “Using Motion Primitives to Generate Initial Guesses for Trajectories to Sun-Earth L_2 ,” in *AAS/AIAA Astrodynamics Specialist Conference*, Boston, MA, Aug. 2025.

- [23] J. S. Parker and R. L. Anderson, “Low-Energy Lunar Trajectory Design,” Hoboken, New Jersey: John Wiley & Sons, 2014.
- [24] N. Bosanac, “Clustering Natural Trajectories in the Earth-Moon Circular Restricted Three-Body Problem,” Under Review, 2025.
- [25] A. Bodin, N. Bosanac, C. Gillespie, “Long-Term Spacecraft Trajectory Prediction Using Behavioral Motion Primitives,” AAS/AIAA Astrodynamics Specialist Conference, August 2025, Boston, MA.
- [26] K. Bruchko and N. Bosanac, “Rapid Trajectory Design in Multi-Body Systems Using Sampling-Based Kinodynamic Planning,” The Journal of the Astronautical Sciences, vol. 72, no. 33, 2025.
- [27] E. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” Numerische Mathematik, vol. 1, pp. 269–271, 1959.
- [28] J. Yen, “Finding the k Shortest Loopless Paths in a Network,” Management Science, vol. 17, pp. 712–716, 1971.