

## DESIGNING NEPTUNIAN SYSTEM TOURS VIA A MOTION PRIMITIVE APPROACH

Giuliana E. Miceli\* and Natasha Bosanac†

Neptune is an important target for future space missions. However, the chaotic dynamics of its multi-body gravitational system render designing complex and constrained tours a challenging task. To address this challenge, a motion primitive approach is used to design tours of the Neptunian system that visit multiple science orbits. First, motion primitives are generated to summarize spatial motion in the Neptune-Triton circular restricted three-body problem. Then, motion primitive graphs are constructed to identify geometrically distinct and constrained trajectories within each possible leg of a tour. These trajectories are sequenced to construct a Neptunian system tour.

### INTRODUCTION

Having only been visited by the Voyager 2 spacecraft in 1989,<sup>1</sup> Neptune is an interesting target for future planetary science missions.<sup>2</sup> Neptune consists mostly of oxygen and carbon,<sup>3</sup> resembling what is likely to be the most common type of planet across the universe and, potentially, exoplanetary systems. The Neptunian system also consists of 16 known moons.<sup>4,5</sup> Triton, the largest moon, follows a retrograde orbit relative to the rotation of Neptune and is postulated to be a captured Kuiper belt object.<sup>6</sup> Because of this expected origin, the composition, atmosphere, and dynamical environment of Triton are also of much interest to the scientific community.<sup>2</sup>

Designing a mission for one or more spacecraft to visit the Neptunian system relies on the design of complex trajectories. Currently, interplanetary transfers to the Neptunian system often produce high values of  $v_\infty$  while the spacecraft possesses limited maneuvering capability. Furthermore, the solution space in the multi-body Neptunian gravitational environment is complex to analyze using traditional dynamical systems techniques. As a result, designing Neptunian system tours while satisfying hardware and mission constraints can be challenging.

Researchers have previously designed trajectories to support mission concepts focused on touring the Neptunian system. Using a traditional patched conics analysis and the two-body problem, Campagnola et al. presented an example of a 2-year tour that begins with a 2.5 km/s Neptunian orbit insertion (NOI) maneuver and requires no deterministic maneuvers thereafter; this tour includes 55 flybys of Triton.<sup>7</sup> To further explore the solution space, Melman et al. used a Neptune-Triton Circular Restricted 3-Body (CR3BP) problem when designing trajectories for a Triton orbiter.<sup>8</sup> More recently, Miceli, Bosanac, Stuart, and Alibay used a motion primitive approach to extract arcs of

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

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

distinct geometries along fundamental solutions in the Neptune-Triton CR3BP. These motion primitives were then used to design planar trajectories of distinct geometries from NOI to a specified science orbit with close passes to Triton.<sup>9</sup> Miceli, Bosanac, and Karimi then extended this work to design constrained, planar trajectories and further explore the trajectory trade space by varying the arrival condition and target science orbit.<sup>10</sup>

Inspired by their extensive use in robotics, a motion primitive approach to spacecraft trajectory design was first introduced by Smith and Bosanac<sup>11,12</sup> and later extended by Miceli et al.<sup>9,10</sup> In robotics, motion primitives represent fundamental building blocks of motion that can be chained together to create a more complex trajectory. To apply this concept to trajectory design, clustering is used to automatically extract arcs of distinct geometries from a specified set of trajectories, supporting the construction of a motion primitive library;<sup>11</sup> in our prior and current work, these motion primitives have summarized arcs along fundamental solutions. The potential connectivity of these motion primitives is then represented by a graph: each node is a state along a primitive, any edges indicate pairs of primitives that may be sequentially composed, and the edges weights capture the differences in position and velocity direction between two primitives.<sup>10</sup> Where necessary, constraints are incorporated into the graph by updating the nodes and edges.<sup>10</sup> This motion primitive graph is searched to obtain primitive sequences that minimize the sum of edge weights.<sup>12</sup> In this work, a custom k-best path algorithm enables the recovery of multiple motion primitive sequences that are then refined to produce geometrically distinct initial guesses.<sup>10</sup> These initial guesses are corrected and optimized to reduce the total  $\Delta V$  and satisfy the specified constraints. Where appropriate, these trajectories are corrected in a higher-fidelity ephemeris model.<sup>12</sup>

Building upon our prior work, this paper uses a motion primitive approach to design tours of the Neptunian system and explore the associated trajectory trade space. Each tour is generated by chaining together several smaller motion primitive sequences that each connect unique combinations of science orbits and/or boundary conditions. Accordingly, each of these sequences is generated by searching smaller subgraphs for each leg of the tour. Independently searching these subgraphs produces multiple, unique motion primitive sequences that are used to recover geometrically distinct trajectories for each leg of the tour. These trajectories are analyzed and sequentially composed to rapidly generate geometrically distinct tours of the Neptunian system. These results demonstrate the value of a motion primitive approach to rapidly exploring a complex trajectory trade space. Furthermore, the tours may support the science community during the development of the mission concept by offering a broad summary of trajectories that may achieve a variety of science goals.

## BACKGROUND

### Circular Restricted Three-Body Problem

The Circular Restricted Three-Body Problem is used to approximate the dynamics governing a spacecraft in the Neptune-Triton system, abbreviated as NT-CR3BP throughout this paper. In this model, Neptune is the first primary body  $P_1$ , Triton is the second primary body  $P_2$ , and the spacecraft is the third body. Furthermore, the spacecraft is assumed to possess a negligible mass with respect to the two primaries. Finally,  $P_1$  and  $P_2$  are modeled as point masses, moving in circular orbits about their barycenter. Given the eccentricity of Triton's orbit around Neptune,  $e_T = 0.000016$ ,<sup>4</sup> this assumption is reasonable.

Trajectories are generated and analyzed using nondimensional coordinates in a Neptune-Triton rotating frame. This frame is defined with the origin at the Neptune-Triton barycenter and axes

$\hat{x}, \hat{y}, \hat{z}$ :  $\hat{x}$  is directed from  $P_1$  to  $P_2$ ,  $\hat{z}$  is in the direction of the orbital angular momentum vector of the primary bodies, and  $\hat{y}$  completes the right-handed triad.<sup>13</sup> In addition, quantities are nondimensionalized using three characteristic quantities  $l^*$ ,  $m^*$ , and  $t^*$ .<sup>13</sup> In the Neptune-Triton system,  $m^* \approx 1.024569 \times 10^{26}$  kg is the sum of the masses of the primaries,  $l^* = 354,760$  km is set equal to the average distance between the primaries, and  $t^* \approx 8.081353 \times 10^4$  sets the mean motion of the primary system to unity.

The equations of motion governing a spacecraft in the CR3BP are written in nondimensional coordinates in the rotating frame. These second-order differential equations are expressed as

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

with

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

where  $r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$  and  $r_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$ . In this model,  $\mu = M_2/(M_1 + M_2) \approx 0.00020895$  is the mass ratio of the Neptune-Triton system. One integral of motion, the Jacobi constant, exists and is equal to  $C_J = 2U^* - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$ .<sup>13</sup>

Although the equations of motion of the CR3BP do not have an analytical solution, fundamental solutions exist in this autonomous system. For instance, five Lagrange or equilibrium points exist in the rotating frame, labeled  $L_i$  where  $i = [1, 5]$ . Additionally, periodic orbits repeat their path in the rotating frame after a minimal time labeled the period and exist in continuous families.<sup>13</sup> Some periodic orbit families emanate from or exist near the equilibrium points or primary bodies. In addition, a  $p : q$  resonant orbit in the CR3BP completes  $p$  revolutions around the larger primary body in approximately the time that the smaller primary body completes  $q$  revolutions in the inertial frame;<sup>14</sup> the resonant orbit is labeled as interior if  $p > q$  and exterior when  $q > p$ . Furthermore, hyperbolic invariant manifolds capture trajectories that approach (stable) or depart (unstable) from unstable periodic orbits asymptotically in time.

## Ephemeris Model

A point mass ephemeris model of Neptune and its moons offers a higher fidelity representation of the dynamical environment. The spacecraft's state is defined in an inertial frame with its origin at the center of Neptune and the axes of the International Celestial Reference Frame. The state of the spacecraft relative to Neptune in the inertial frame is defined as:

$$\bar{X}_{N,sc} = [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}]^T = [\bar{R}_{N,sc}, \bar{V}_{N,sc}]^T \quad (3)$$

where the subscripts  $N$  and  $sc$  indicate Neptune and the spacecraft. Furthermore, the spacecraft is assumed to possess a negligible mass in comparison to the mass of the primaries. With this foundation, the following equations of motion govern the path of the spacecraft:

$$\ddot{\bar{R}}_{N,sc} = -GM_N \left( \frac{\bar{R}_{N,sc}}{R_{N,sc}^3} \right) + G \sum_{i=1}^{N_b} M_i \left( \frac{\bar{R}_{sc,i}}{R_{sc,i}^3} - \frac{\bar{R}_{N,i}}{R_{N,i}^3} \right) \quad (4)$$

where  $M_i$  is the mass of body  $i$ ,  $N_b$  is the number of total bodies,  $G$  is the universal gravitational constant,  $(\dot{\cdot})$  indicates a time derivative with respect to an observer in the inertial frame, and

$\bar{R}_{i,j}$  indicates the position vector measured from body  $i$  to body  $j$ . To evaluate these equations, ephemerides locating Neptune and its 14 moons are extracted from the following kernels provided by NASA's Navigation and Ancillary Information Facility (NAIF):<sup>15,16</sup> DE440, nep097, nep095 and nep102.<sup>17</sup> Additionally, the naif0012 file is used for time calculations whereas the pck00011 kernel is used for frame transformations.

### Numerically Correcting Trajectories

Coarsely designed initial guesses are corrected and optimized in the Neptune-Triton CR3BP and then the ephemeris model by using collocation. In this paper, the formulation of the collocation problem follows the approach presented by Grebow and Pavlak<sup>18</sup> and then used in motion primitive-based trajectory design by Smith and Bosanac.<sup>12</sup> Collocation relies on fitting polynomials of  $N$ -th order using states distributed along arcs of a trajectory. The polynomials are constrained to closely approximate solutions to a specified dynamical model by minimizing the residuals between the time derivatives of the polynomial and the evaluated dynamics at other nodes along each arc. These variables and constraints are mathematically encoded into a free variable-constraint vector formulation of the corrections problem. As a result, the polynomials are updated iteratively via Newton's method until continuity and dynamical constraints are satisfied to within a specified tolerance.

The first step of collocation involves creating a mesh. Each initial guess is discretized into  $m$  segments. Then, each segment is split into  $n$  arcs, and  $p$  nodes are placed along each arc. Although the number and spacing of segments and arcs are defined by the trajectory designer, the number and location of nodes are defined by the order and type of collocation scheme employed. The approach implemented in this paper leverages a 7-th order polynomial collocation scheme with a Legendre-Gauss-Lobatto (LGL) node spacing strategy. Thus, each arc is discretized into  $p = 7$  nodes, placed at normalized times  $\tau \in [-1, 1]$  that are equal to the roots of the derivative of the  $(p - 1)$ -th order Legendre polynomial.<sup>18-20</sup> The odd-numbered nodes, i.e.,  $k = 1, 3, 5, 7$ , are used to fit the polynomial  $q(x)$  along the arc and, therefore, are labeled as free nodes. However, the even-numbered nodes, i.e.,  $k = 2, 4, 6$ , are used to assess the differences between the polynomial representation and the system dynamics; these nodes are denoted as defect nodes.

A free variable vector includes the states and integration times used to generate each polynomial. The state at the  $k$ -th node of the  $j$ -th arc along the  $i$ -th segment of the trajectory is described by the state vector  $\mathbf{x}_{j,k}^i$  whereas  $\Delta t_j^i$  is the integration time along the arc. Then, the free variable vector  $\mathbf{V}_i$  for the  $i$ -th segment is expressed as

$$\mathbf{V}_i = \begin{bmatrix} \begin{bmatrix} \mathbf{x}_{1,1}^i \\ \mathbf{x}_{1,3}^i \\ \mathbf{x}_{1,5}^i \end{bmatrix}^T & \begin{bmatrix} \mathbf{x}_{2,1}^i \\ \mathbf{x}_{2,3}^i \\ \mathbf{x}_{2,5}^i \end{bmatrix}^T & \dots & \begin{bmatrix} \mathbf{x}_{n_i-1,1}^i \\ \mathbf{x}_{n_i-1,3}^i \\ \mathbf{x}_{n_i-1,5}^i \end{bmatrix}^T & \begin{bmatrix} \mathbf{x}_{n_i,1}^i \\ \mathbf{x}_{n_i,3}^i \\ \mathbf{x}_{n_i,5}^i \\ \mathbf{x}_{n_i,7}^i \end{bmatrix}^T & \begin{bmatrix} \Delta t_1^i \\ \Delta t_2^i \\ \vdots \\ \Delta t_{n_i}^i \end{bmatrix}^T \end{bmatrix} \quad (5)$$

Note that  $\mathbf{x}_{j,7}^i$  is not included in  $\mathbf{V}_i$  because the last node of the  $j$ -th arc coincides with the first node of the  $j + 1$ -th arc. The free variable vector for the entire trajectory  $\mathbf{V}$  is then formed by the free variable vectors  $\mathbf{V}_i$  for all  $m$  segments, resulting in a total of  $((3p - 2) \sum_{i=1}^m n_i + 6m)$  variables.

A constraint vector is defined to enforce continuity at the free nodes and the dynamics at the defect nodes. The continuity constraint vector along the  $i$ -th arc is defined as

$$\mathbf{F}_c^i = \begin{cases} (\mathbf{x}_{1,1}^{i+1} - \mathbf{x}_{n_i,p}^i)^T & \text{if natural motion} \\ (\mathbf{r}_{1,1}^{i+1} - \mathbf{r}_{n_i,n}^i)^T & \text{if impulsive maneuver applied} \end{cases} \quad (6)$$

for  $i < m$ . Then, the defect constraint vector enforcing the system dynamics at the defect nodes along the  $j$ -th arc of the  $i$ -th segment is defined as

$$\mathbf{F}_{d,j}^i = \begin{bmatrix} (\dot{\mathbf{q}}_{j,2}^i(\tau_2) - \dot{\mathbf{x}}_{j,2}^i)w_2 \\ (\dot{\mathbf{q}}_{j,4}^i(\tau_4) - \dot{\mathbf{x}}_{j,4}^i)w_4 \\ (\dot{\mathbf{q}}_{j,6}^i(\tau_6) - \dot{\mathbf{x}}_{j,6}^i)w_6 \end{bmatrix}^T \quad (7)$$

where  $w_k$  is the LGL weight associated with the  $k$ -th collocation node,  $\dot{\mathbf{q}}$  is the derivative of the polynomial along the arc with respect to normalized time  $\tau$ , defined as  $\tau = 2((t_{j,k}^i - t_{j,1}^i)/\Delta t_j^i) - 1 \in [-1, 1]$ , and  $t_{j,k}^i$  is the time at the  $k$ -th node along the  $j$ -th arc and  $i$ -th segment. In this expression,  $\dot{\mathbf{x}}$  is the normalized time derivative of the state vector  $\mathbf{x}_{j,k}^i$  calculated as

$$\dot{\mathbf{x}}_{j,k}^i = \frac{\Delta t_j^i}{2} \mathbf{g}(\mathbf{x}_{j,k}^i) \quad (8)$$

where  $\mathbf{g} = [\dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]$ . For all  $n_i$  arcs along the  $i$ -th segment, the defect constraint vector is

$$\mathbf{F}_d^i = [\mathbf{F}_{d,1}^i, \mathbf{F}_{d,2}^i, \dots, \mathbf{F}_{d,n_i}^i] \quad (9)$$

Finally, the constraint vector  $\mathbf{F}(\mathbf{V})$  for the entire trajectory is defined as

$$\mathbf{F}(\mathbf{V}) = [\mathbf{F}_c^1, \mathbf{F}_c^2, \dots, \mathbf{F}_c^{m-1}, \mathbf{F}_d^1, \mathbf{F}_d^2, \dots, \mathbf{F}_d^m]^T \quad (10)$$

Once the vectors  $\mathbf{V}$  and  $\mathbf{F}$  are computed, Newton's method is used to iteratively update  $\mathbf{V}$  until the norm of the constraint vector is equal to zero within a tolerance of  $10^{-12}$  in the NT-CR3BP.

Following correction, the mesh that is composed of a set of segments, arcs, and nodes is refined to improve the accuracy of the polynomial approximation of the trajectory. This refinement step enables arcs to possess different lengths in more or less sensitive regions of the phase space. The mesh refinement step aims to equally distribute the error on the constraint nodes along the arcs of the solution.<sup>21</sup> In this paper, hybrid mesh refinement is implemented using the approach presented by Grebow and Pavlak and the method for error redistribution by Carl de Boor;<sup>18,22,23</sup> this approach was previously implemented by Smith and Bosanac for correcting primitive-based initial guesses.<sup>12</sup>

### Constrained Local Optimization

Each corrected trajectory is optimized to minimize the maneuver cost while maintaining geometric similarity to the initial guess. Optimization is implemented by embedding the collocation corrections problem inside a direct optimization scheme using MATLAB's `fmincon` function with the `sqp` solver.<sup>24,25</sup> In this optimization problem, a multi-objective cost function is defined in terms of the free variable vector  $\mathbf{V}$  as

$$J(\mathbf{V}) = w_{geo}(\Delta \mathbf{r}_{IG-CG} \cdot \Delta \mathbf{r}_{IG-CG}) + w_{man} \sum_{i=1}^{n_m} (\Delta v_i)^2 \quad (11)$$

In this expression,  $\Delta \mathbf{r}_{IG-CG}$  is the difference between the position vectors of each collocation node along the initial guess and current guess, and  $\Delta v_i$  is the magnitude of the  $i$ -th of  $n_m$  impulsive maneuvers along the trajectory. The two coefficients,  $w_{geo}$  and  $w_{man}$ , scale the geometric and maneuver components of the objective function, respectively. The values of these weights are varied

via continuation, starting from  $[w_{geo}, w_{man}] = [1, 0]$  to prioritize geometric similarity to the initial guess until reaching  $[w_{geo}, w_{man}] = [0, 1]$  to prioritize minimizing maneuver requirements in a user-defined number of steps.

During each step of the optimization process, the solution must also satisfy several constraints. First, the equality constraints encoded in  $\mathbf{F}$  are used to enforce continuity between arcs and dynamical accuracy. Additional user-defined constraints are included as inequality constraints, including maximum time of flight (TOF), maximum total maneuver magnitude  $\Delta v_{tot}$ , maximum or minimum magnitude of a single maneuver  $\Delta v$ , and maximum or minimum distance from Neptune or Triton. The gradients of these constraints with respect to the free variables are analytically derived to ensure accuracy and reduce computational time.

## TECHNICAL APPROACH

The motion primitive approach to spacecraft trajectory design is composed of five steps, as presented previously by Smith and Bosanac and then Miceli and Bosanac.<sup>9,10</sup> These steps include:

1. Extract motion primitives from a specified set of fundamental solutions.
2. Generate a motion primitive graph to estimate of the primitives' sequential composability.
3. Search the graph to obtain primitive sequences that minimize cumulative state discontinuities.
4. Refine each sequence of motion primitives to produce geometrically distinct initial guesses.
5. Correct the refined initial guess with impulsive maneuvers in an ephemeris model.

Although this paper leverages prior work as a foundation, several substantial improvements appear in this paper. First, the approach used to extract motion primitives has been updated to improve their quality and accuracy. Specifically, trajectories sampled along fundamental solutions are sampled using curvature maxima to supply a consistent definition. In addition, a hierarchical density-based algorithm is used to extract groups of sufficiently similar arcs prior to primitive extraction.<sup>26</sup> In addition, this paper presents a new formulation of a motion primitive graph that better estimates the sequential composability of primitives. Furthermore, a new search algorithm is presented to produce a wider array of primitive sequences that results in a more diverse set of initial guesses. This diversity supports better exploring the trajectory trade space. Finally, the refinement process has been improved to produce higher-quality initial guesses.

### Step 1: Motion Primitive Generation

Motion primitives supply fundamental building blocks of motion.<sup>27</sup> In this paper, motion primitives are generated to summarize arcs along periodic libration points and resonant orbits as well as hyperbolic invariant manifolds in the NT-CR3BP.<sup>10</sup> These fundamental solutions are segmented into arcs in a geometry-based manner. Then, these arcs are clustered to discover geometrically similar groups. One representative member of each cluster serves as the motion primitive.

To define the arcs used to form motion primitives, the concept of curvature is employed.<sup>28</sup> At any state along a trajectory, the curvature  $\kappa$  is calculated as

$$\kappa(t) = \frac{\sqrt{(\ddot{z}\dot{y} - \ddot{y}\dot{z})^2 + (\ddot{x}\dot{z} - \ddot{z}\dot{x})^2 + (\ddot{y}\dot{x} - \ddot{x}\dot{y})^2}}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}} \quad (12)$$

Thus, maxima in the curvature satisfy the following condition:  $\dot{\kappa} = 0$  and  $\ddot{\kappa} < 0$ . These maxima often occur in similar locations to apsides relative to meaningful reference points, enabling a consistent definition of primitives throughout a multi-body system.

Trajectories that lie along hyperbolic invariant manifolds are segmented into smaller arcs, each defined to include a specified number of maxima in the curvature. First, trajectories are generated along the selected stable or unstable manifold. Then, initial conditions for arcs are selected at each maximum in the curvature along these trajectories. Each arc is then defined to span an additional  $n_{\kappa max} = 4$  maxima.<sup>10</sup> Periodic and resonant orbit families, however, are used in their entirety, since their geometry is repeated periodically they are distinct from other orbits in their integrity.

Each arc is described using a finite-dimensional feature vector. In this work, each arc is discretized into  $n_L$  states that are equally distributed along the arclength. Then, the feature vector is composed of the position vectors at each sampled state as

$$\mathbf{f} = [x_1, y_1, z_1, \dots, x_{n_L}, y_{n_L}, z_{n_L}]^T \quad (13)$$

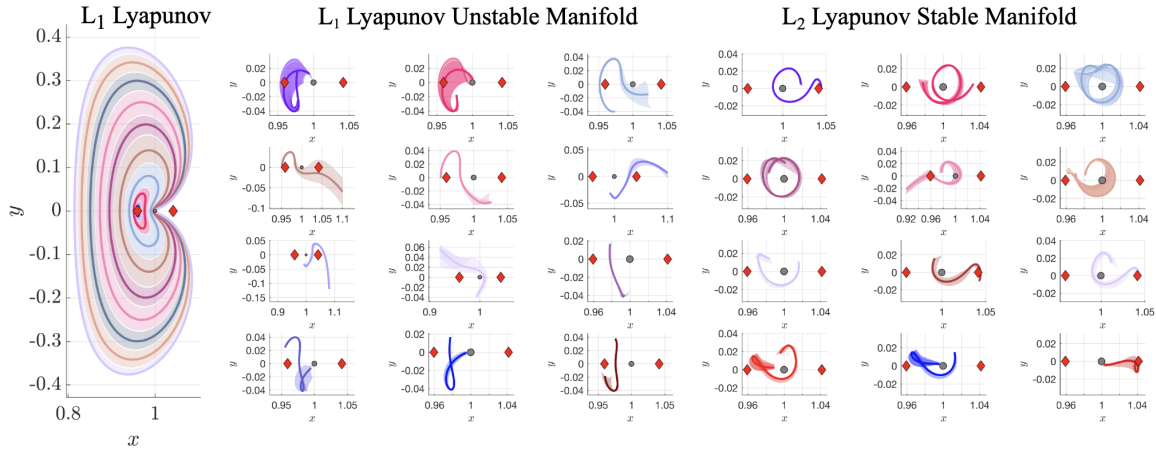
to produce a  $3n_L$ -dimensional description. In this paper,  $n_L = 15$  is selected to balance the level of detail in the description with limiting the dimensionality of the feature vector.

The feature vectors describing the arcs along each fundamental solution are clustered to identify groups of geometrically similar trajectories. This clustering step is performed with the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN),<sup>26</sup> an algorithm that has been successfully employed in a variety of astrodynamics applications to group trajectories by their geometry.<sup>28–33</sup> HDBSCAN discovers clusters as members with a sufficient density in a feature vector space, with the selected clusters also maximizing the stability across a hierarchy of all possible clusters. This algorithm is used in this paper because it does not require a priori knowledge of the number of clusters and it can detect clusters of different densities and shapes.<sup>26</sup>

The inputs to HDBSCAN are the feature vectors describing a set of trajectories and two hyperparameters,  $n_{min,size}$  and  $n_{min,core}$ . The first hyperparameter,  $n_{min,size}$ , specifies the minimum number of members in a cluster whereas  $n_{min,core}$  defines the size of the neighborhood used to calculate density.<sup>26</sup> In this paper, the values of  $n_{min,core} = 5$  and  $n_{min,size} = 15$  are used to cluster the majority of the periodic orbit families. The output of HDBSCAN is the cluster assignment of each arc. If an arc has not been assigned to any cluster, it is designated as noise.

Noise elements are not necessarily members that do not resemble other members in the datasets. In many cases, noise points are trajectories existing in a group smaller than the defined  $n_{min,size}$ , or existing at the boundary of very dense clusters, that the algorithm can fail to recognize due to the specific user-defined inputs. The number of noise points obtained after clustering each trajectory dataset is minimized by applying two noise mitigation steps.<sup>10</sup> First, the noise points are compared to the  $n_{min,core}$  elements of each cluster to decide if they should be merged within a cluster, removing the noise existing at the boundary of each cluster. Then, if other noise trajectories are left, these are clustered in a second clustering step with different values of  $n_{min,core}$  and  $n_{min,size}$  to obtain those smaller clusters that were not identified in the previous step. The values of  $n_{min,core} = 1$  and  $n_{min,size} = 2$  are used in this step. The combination of these noise mitigation steps usually relabel most noise points. However, if the second clustering outputs noise points, these are discarded from the dataset as they can be considered isolated elements that do not represent a fundamental type of motion.<sup>10</sup> An example of the geometrically different clusters obtained with this approach is shown in Figure 1, both for periodic orbits such as a  $L_1$  Lyapunov family and for stable and unstable invariant manifolds.

Finally, a motion primitive is extracted from each cluster as the medoid. This member of a cluster possesses the minimum cumulative distance from all the other members, as presented in the original



**Figure 1. Selected motion primitives (thick curves) and their region of existence (shadowed area) for the  $L_1$  Lyapunov orbit family, the unstable manifold of an  $L_1$  Lyapunov orbit at  $C_J = 3.00092$ , and the stable manifold of an  $L_2$  Lyapunov orbit at  $C_J = 3.01383$**

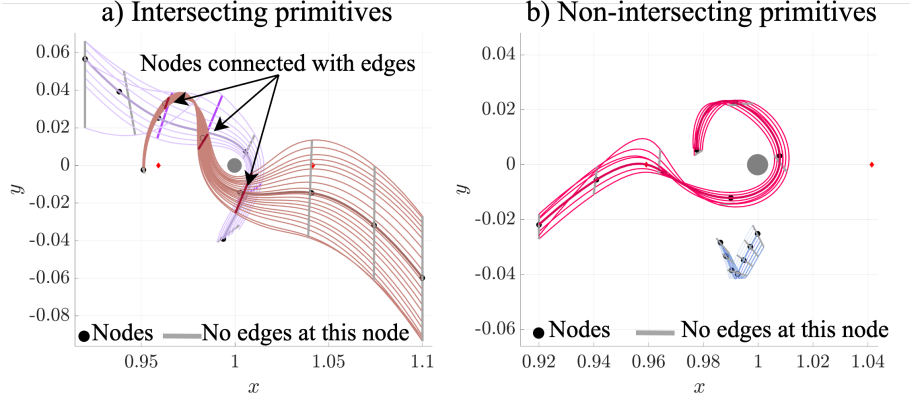
implementation of the motion primitive approach by Smith and Bosanac.<sup>12</sup> Then within the cluster,  $k$  trajectories at an equal distance from the primitive are selected to obtain a representation of the region of existence (*roe*) of that motion type in the solution space. In Figure 1, the primitives are highlighted with thick lines, while the light-colored region around them represents the region of existence. The primitives and their regions of existence are stored in a motion primitive library; this library offers both a condensed, discrete representation of the continuous solution space and supports constructing complex trajectories.

## Step 2: Motion Primitive Graph

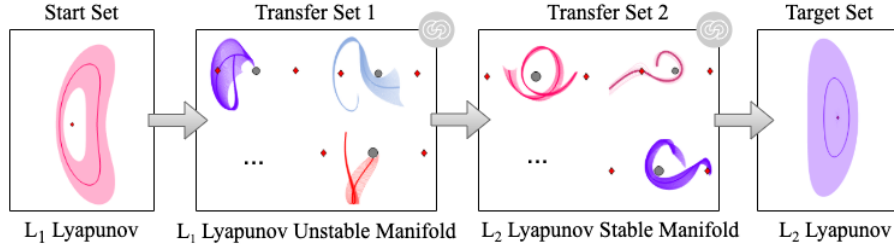
Graphs have been used in robotics and other disciplines to represent relationships between elements in a given space, such as locations on a map for a path-planning problem.<sup>34</sup> These structures are composed of nodes and edges. The edges can be directional or non-directional and weighted or unweighted, depending on the type of relationships represented. In previous applications of path-planning with motion primitives, both in robotics and astrodynamics, graphs have been formulated to assign distinct motion primitives as nodes and directed, weighted edges to describe the connectivity between the primitives.<sup>9, 12, 32, 35, 36</sup> As a result, motion primitive graphs can supply a discrete summary of segments of a continuous solution space.

A motion primitive graph is formulated to capture the potential sequential composability of motion primitives in the NT-CR3BP. The nodes are defined to correspond to states along each motion primitive. For instance, each primitive is sampled by  $n_s$  states, evenly distributed along the arclength, to contribute  $n_s$  nodes to the graph. Accordingly, a graph capturing  $m_p$  motion primitives includes  $m_p n_s$  nodes. Then, edges are added only between nodes if the regions of existence of the two primitives overlap in position near those nodes and if any additional requirements are satisfied. An example of this edge construction step is depicted conceptually in Figure 2. For a given node  $n_a$ , if the span of the roe at that node is large enough to reach another node  $n_b$  along a different primitive, then an edge is created between  $n_a$  and  $n_b$ , otherwise no edge is created between those nodes. Compared to our prior work, this motion primitive graph better captures potential connectivity between two primitives and adaptively adds edges.





**Figure 2.** Example of: a) intersecting primitive's region of existence leading to edges placement between nodes in the graph; and b) non-intersecting primitive's regions that do not produce edges between nodes in the graph.



**Figure 3.** Example of high-level itinerary graph for designing a transfer between an  $L_1$  Lyapunov orbit at  $C_J = 3.0092$  and  $L_2$  Lyapunov orbit at  $C_J = 3.01383$ , using the primitives from their unstable and stable manifolds in sequence. The gray symbol at the top right corner of the transfer set boxes indicates that the primitives in that set can also be interconnected.

Edges are added based on a high-level itinerary graph, which is defined by the trajectory designer at the beginning of this step as presented by Smith and Bosanac.<sup>12</sup> Here, the designer defines the groups of primitives of interest from the database computed in Step 1. Then, the designer assigns them to a block defining the order in which they can be explored during the path search. An example of a high-level itinerary graph is provided in Figure 3. Here, the primitive in one block can be connected with the primitives in the next block on the right. Moreover, the gray symbol at the top right of the transfer set blocks indicates that the primitives within the set can all be interconnected. This itinerary graph can reflect a trajectory designer's expertise in a scenario or constrain the itinerary of a trajectory.

As presented in previous work,<sup>9,10,12</sup> the edge weights estimate the potential sequential composability of motion primitives that begin and end at the selected pair of nodes. These edges are weighted by the discontinuity in position  $\Delta r$  and velocity direction  $\Delta\alpha(\vec{v})$  between two states along one or two primitives, i.e., two nodes  $i$  and  $j$  in the graph. This edge weight is defined as<sup>10</sup>

$$q_{i,j} = \Delta r_{i,j} + (1 - \Delta\alpha(\vec{v}_{i,j})) \quad (14)$$

Note, however, that the nodes belonging to the same primitives are connected with  $q = 0$ .

The graph construction step also incorporates additional constraints, such as minimum or maxi-

imum distance from a body and maximum single maneuver cost.<sup>10</sup> The constraints on distance from a primary body act on the primitives and their regions of existence. In particular, if any primitive or any trajectory in the region of existence violates the distance threshold, the trajectory is flagged for removal from the set. If all the trajectories in the region of existence associated with a primitive are flagged, including the primitive, then that entire primitive set is removed from the graph. Otherwise, if only some trajectories are flagged, the region of existence is simply resized to contain only the members satisfying the constraints. In the case that the primitive of the set is removed, another primitive is identified among the remaining trajectories. On the other hand, constraints on maneuver cost modify the edges. When the edge weight between two nodes is computed, the estimated maneuver cost is evaluated as  $\Delta \vec{v}_{i,j} = |\vec{v}_j - \vec{v}_i|$ . If this cost is higher than the specified constraint value for a single maneuver, then that edge is removed from the graph.

Any constraint defined at this step incorporates a margin from the actual desired value, as the graph represents a discrete approximation of the solution space. For example, if the mission requires a minimum allowable altitude from the surface Triton of  $h_{min} = 500 \text{ km}$ , this constraint can be imposed on the graph step as  $h_{min} = 300 \text{ km}$ . Then, the initial guesses resulting from a constrained graph can be constrained during corrections using the actual value of 500 km. The constraints at this step of the process can be considered as a way to filter the solution space.

### Step 3: Generate Sequences of Motion Primitives

Consistent with path-planning literature, motion primitives can be composed in sequences to form complex paths.<sup>27,37</sup> Accordingly, consistent with the approach presented by Smith and Bosanac,<sup>12</sup> the motion primitive graph is searched to produce motion primitive sequences; these sequences support the initial guess construction process. However, this step presents multiple challenges due to the complexity of discovering a diverse array of primitive sequences with the lowest cumulative state differences in a computationally feasible manner. Accordingly, this paper presents a new custom algorithm to search the motion primitive graph. This algorithm combines two well-known path planning algorithms, A\* and Yen's algorithm,<sup>38,39</sup> with some modifications to ensure that the resulting paths minimize the total edge weights and exhibit sufficient diversity.

Yen's algorithm is used to identify  $k$  loopless paths within a directed weighted graph.<sup>39</sup> This algorithm works iteratively: 1) at each iteration of the algorithm, the best solution within the graph is identified through Dijkstra's algorithm, and saved in the output list (or list A); 2) the graph is modified by iteratively removing one edge from the best path at each iteration from the original graph; and 3) Dijkstra's algorithm is used to identify paths within these subgraphs<sup>40</sup> and stores them in a temporary list, or list B. The best solution of the temporary list B is saved in list A. This process is repeated until the  $k$  best of all possible paths are identified.<sup>39</sup>

In this work, A\* is used instead of Dijkstra's algorithm to identify the minimum cost path at each iteration.<sup>38</sup> This algorithm is preferred because it uses a heuristic function to guide the exploration of nodes in the graph, making the search more efficient. Starting at node  $i$  at the current iteration, A\* explores the neighboring nodes associating to each of them the cost  $c_{i,j} = g_{i,j} + h_j$ , where  $g_{i,j}$  is the cost to go from the current node  $i$  to the next node  $j$ , and  $h_j$  is the expected cost to go from the next node  $j$  to the target node.<sup>38</sup> All the combinations of nodes  $N_s$  with  $s = 1, \dots, i$  and the neighboring nodes  $j$  are stored in a queue as incomplete paths. Then, the incomplete path that has the lowest cost of  $\sum c_{i,j} = c_{tot}$  is evaluated for expansion at the next iteration. This process is repeated, computing the cost at the final node of the best incomplete path until the target node is reached. The first path that reaches the target node is the path with the minimum cost over the entire

graph.

When using A\*, the heuristic cost can be selected based on the graph characteristics. However, the heuristic must satisfy the condition  $h_j \leq h_j^*$ , where  $h^*(n)$  is the true or optimal cost to go from the current node to the target node, to ensure that A\* can always return the least expensive path. In this work, the heuristic is selected to equal the discontinuity in position between two nodes  $i$  and  $j$ , which ensures that the relationship  $h_j \leq h_j^*$  is always respected:  $h_j = \sum_{s=j+1}^T \Delta r_{j,s}$  and  $h_j^* = \sum_{s=j+1}^T q_{j,s}$ , where  $q_{j,s}$  is the edge weight defined in Eq. 14 and  $T$  is the index indicating the target node.

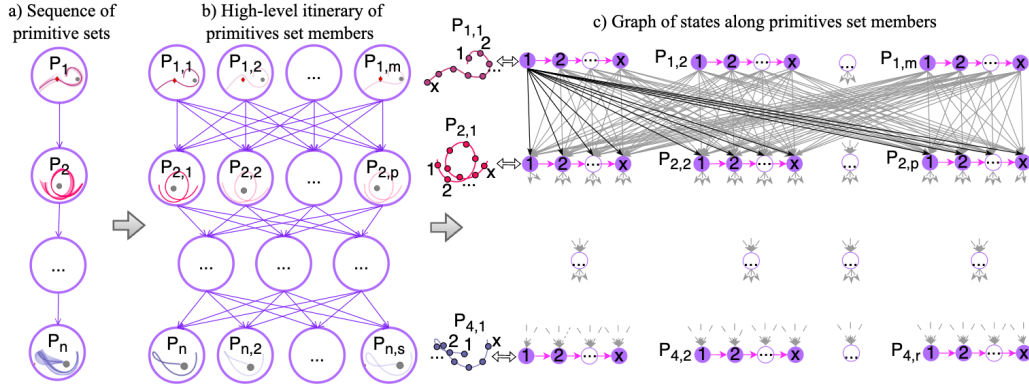
The combination of Yen's algorithm and A\* is used to obtain a user-defined number of diverse paths. Generally,  $k$ -shortest paths algorithms such as Yen's algorithm are formulated to identify the  $k$  paths with the lowest cost, i.e., those that have the closest cost to the best overall path. Within a reasonable number of iterations, this approach leads to paths that are very similar to each other and often closely resemble the best overall path. These solutions have limited interest in many applications<sup>41</sup> and do not give a sufficient representation of the solution space that is summarized by the graph. Consequently, in this paper, the search algorithm is modified to ensure that a sufficiently diverse set of paths is discovered using a wider variety of motion primitives.

The first modification to the search algorithm is the composition of a queue formed by A\*, and in particular, during the evaluation of the neighbors of a node  $i$ . After the first overall best path is found, Yen's algorithm is applied to remove pieces of the best path. Then, A\* is used again on the modified graph that does not include the edges to the removed parts of the best path. During this iteration of A\*, the neighboring nodes  $j$  to the current node  $i$  are ranked by cost. If the node  $i$  belongs to the first primitive associated with a boundary condition, the only expanded neighbor  $j$  is selected to be a random member among its  $k = 10$  nearest neighbors. This modification encourages the diversification of early primitives in a sequence, which has been observed to support the exploration of the solution space. Otherwise, if node  $i$  belongs to any other primitives, all  $k = 10$  nearest neighbors are expanded where these can or cannot randomly include the first neighbor to improve diversification without discarding the globally optimal path. This logic is summarized as

$$\begin{cases} \text{if } n_i \in P_1 \rightarrow n_j = \text{rand}(k_{10}) + 1; \\ \text{if } n_i \in P_X, X = [2, N_p] \rightarrow n_j = k_{10} + \text{rand}(0, 1); \end{cases} \quad \text{where } k_{10} = [1, 10] \quad (15)$$

This modification in the node selection enforces the exploration of the neighboring paths within the algorithm's first few iterations. A second modification to the original Yen's algorithm is applied during the iterations to identify the next best path by removing nodes from the last best solution. Instead of removing one node of the last best path at a time, nodes belonging to the same primitive are simultaneously removed at each iteration to ensure distinct primitive sequences are recovered. At the same iteration, the graph is modified accordingly to remove the edges corresponding to that set of nodes.

Finally, all the paths obtained after each modification of the last best solution are stored in the temporary list B. When the iterations are complete, i.e., when A\* was applied to all the subgraphs generated from the last best path and new paths are found, then list B is sorted and its best solution is saved in list A. However, before adding a complete path from the queue to list B, this solution is compared to all the paths in list A and the solutions in the temporary list B to ensure that the new-found path is composed of different sequences of nodes. As a result, the paths returned after searching with the custom algorithm include the best overall path and the closest most diverse



**Figure 4. Graph construction approach for initial guess refinement**

neighboring solutions.

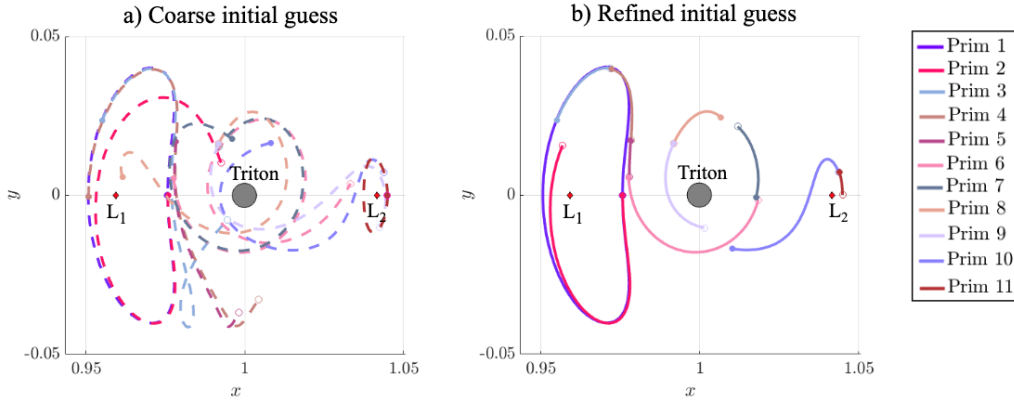
#### Step 4: Initial Guess Refinement

Once the  $k$ -best initial guesses are found, they are refined to obtain smoother, discontinuous sequences of arcs that minimize the state discontinuity. This process is formulated using a second graph, upgrading the original refinement process presented by Smith and Bosanac.<sup>9,10,12</sup> This approach supplies a globally best, refined path.

The refinement process uses both the primitives and its region of existence. This process is composed of three steps as depicted conceptually in Figure 4. First, each initial guess is considered as a sequence of motion primitive sets, where each set is composed of the motion primitive and 20 selected exemplars in its region of existence, as displayed in Figure 4a). Then a high-level graph is constructed using each member of a motion primitive set as a node. In this graph, the nodes of the same set are not connected but every member of a motion primitive set is connected with every member of the subsequent set, allowing every possible combination within the primitives and their exemplars. This step is displayed in Figure 4b). Then, the final graph is built using the states sampled along each trajectory as the nodes. The nodes representing the states along one trajectory are sequentially connected with zero-weight edges to capture natural flow, represented with magenta arrows in Figure 4c). At this stage, the nodes belonging to trajectories in the same sets are not connected. However, every node representing a state along one trajectory in the  $P_i$ -th set is connected via directed edges to all the nodes in the set  $P_{i+1}$ , represented as black and grey arrows in Figure 4c). The edge weights are computed using the difference in position and velocity direction between two nodes, as expressed in Equation 14. Here, the constraints imposed in Step 2 are still enforced.

The completed graph is searched using the A\* algorithm<sup>38</sup> to obtain one path that minimizes the cumulative edge weights. The retrieved path is composed of 1) a subset of nodes along one trajectory associated with each primitive set in the sequence, and 2) edges that reflect either natural coast segments or state discontinuities between these arcs. To mitigate the potential for any primitive sets to be reduced to a single node with no coast segments, the search algorithm is modified to enforce each sequence be composed of at least two elements along the same primitive set, i.e.,  $P_{i,j} = \{p_{i,j,k}\} \geq 2$ .

To demonstrate the initial guess construction process, consider a foundational transfer example



**Figure 5. a) Coarse initial guess obtained from Step 3 and b) Refined initial guess obtained from Step 4**

from an  $L_1$  Lyapunov orbit at  $C_J = 3.0092$  to an  $L_2$  Lyapunov orbit at  $C_J = 3.01383$  in the NT-CR3BP as shown in Figure 3. A coarsely constructed initial guess from Step 3, composed of a sequence of 11 primitives, is displayed in Figure 5a). Following refinement, as described in this section, the resulting initial guess is displayed in Figure 5b). Through this process, the final initial guess minimizes discontinuities and overlap between arcs, such that the solution has a higher likelihood of successful corrections than the original sequence of primitives.

### Step 5: Trajectory Correction and Optimization

The refined initial guesses are first corrected and optimized in the Neptune-Triton CR3BP via collocation. The final continuous and optimal trajectory is then corrected and optimized also in the ephemeris model via collocation. At each step, impulsive maneuvers are allowed at selected locations along the trajectory.

In the first step of the correction process, each initial guess is discretized to form the initial mesh for the collocation. Thus user-defined nodes are placed along the trajectory to create segments and arcs. Then 7 collocation nodes are placed along each arc using the values of  $\tau$  as described in the background section about correction. The set of collocation nodes for each arc along of every segment forms the initial mesh. At this stage, maneuver locations can be placed at any of the nodes of the mesh and at the initial or final state of each primitive.

Using the collocation-based optimization scheme described in the Background section, the initial guess is corrected to minimize maneuver costs while maintaining its geometry. Using continuation, the weights of the objective function are modified in a user-defined number of steps from  $w_{geo} = 1$  and  $w_{man} = 0$ , to  $w_{geo} = 0$  and  $w_{man} = 1$ , as originally presented by Smith and Bosanac.<sup>12</sup> At each step, the trajectory is corrected and optimized. This corrected trajectory seeds the initial guess for the next corrections problem at a new combination of  $w_{geo}$  and  $w_{man}$ . As a result of this approach, the transfer obtained at each step might gradually evolve from the initial guess as the total maneuver magnitude decreases. If desired, path and/or maneuver constraints may be added during the correction and optimization step using the exact values required to meet the mission requirements.

Finally, the optimal transfers in the CR3BP are corrected in the ephemeris model following the same approach that uses collocation and optimization.<sup>9,10</sup> In this implementation, the central body

is Neptune and the secondary bodies are the Neptunian moons: Triton, Naiad, Thalassa, Despina, Galatea, Larissa, Hippocamp, Proteus, Nereid, Halimede, Sao, Laomedeia, Psamathe, Neso. Although these trajectories are corrected in the inertial frame, they are visualized in the Neptune-Triton rotating frame.

## RESULTS

The presented technical approach is used to construct tours of the Neptunian system that support a variety of scientific goals.

### Initial Condition

Each tour begins when the spacecraft arrives in the Neptunian system after completing the interplanetary transfer. To identify an initial condition, a Neptune Orbit Insertion (NOI) state that occurs at the first periapsis with respect to Neptune is used. The initial condition primitives are obtained from the NOI states propagated backward and forward in time for 3.5 days respectively. In this paper, two NOIs with different epochs and energy are used. The first primitive arrives at the Neptunian system on October 2nd, 2045 and it is characterized by  $C_J = 0.950385$ . The second primitive from an NOI arrives on January 11th, 2052, and a Jacobi constant of  $C_J = 0.896031$ . These primitives are plotted in this order in the Start set of Figure 7a). These two arcs supply two possible initial conditions for the first leg of the tour.

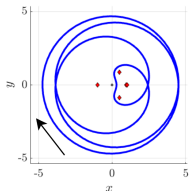
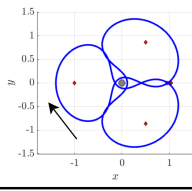
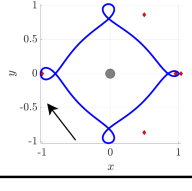
### Target Selection

For this work, the main goals for a tour are obtaining close approaches to the planet and its two main moons, Proteus and Triton. Three science orbits are selected to achieve these goals. These orbits define boundary conditions for each motion primitive subgraph. Note, however, that additional science orbits could be identified and used to extend the tour.

First, a 1:4 exterior resonant orbit with a period of 23.51 days and Jacobi constant of  $C_J = 1.9084$  is selected to support studying Proteus. This orbit is displayed in the first row of the table in Figure 6. This orbit possesses a periapsis with respect to Neptune of 135,761.8969 km, which would place the spacecraft at  $\sim 18,114$  km from Proteus's orbit. With proper timing and suitable instruments, this periapsis distance could support observations of Proteus for analysis of its surface and composition, which are usually challenging given that its dark surface possesses a reflectivity rate of only 6%.<sup>42</sup> Due to its low value of the Jacobi constant, this orbit is selected as the first target along the tours following the initial condition.

The second target of these tours is selected as a 3 : 2 interior resonant orbit to support close passages to Triton. This orbit is displayed in the second row of the table in Figure 6. It has a higher Jacobi constant and lower period than the first science orbit. Given its low periapses with respect to both Neptune and Triton, this orbit could be leveraged to study both celestial bodies from a closer vantage point. Moreover, a spacecraft along this 3 : 2 resonant orbit encounters Triton two times during each orbit period, which allows the analysis of this moon at different phase angles along its orbit around Neptune.<sup>43</sup> This orbit is selected as the second orbit of the tours due to its intermediate value of the Jacobi constant.

Finally, the 4 : 3 interior resonant orbit is selected as the third target of the Neptunian system tours to support studying Triton. This orbit is displayed in the bottom row of Figure 6. The orbit maintains a lower distance from and lower altitude flyby of Neptune's main moon, Triton. This geometry

Target Orbit	Properties	Scientific Goal
<b>1:4 Resonant Orbit</b> 	Period: 23.51 days Periapsis to Triton: 218998.1031 km Periapsis to Neptune: 135761.8969 km Semi-major axis: 1787863.8453 km Jacobi Constant: 1.9084	Close passages to Proteus
<b>3:2 Resonant Orbit</b> 	Period: 11.81 days Periapsis to Triton: 6949.5642 km Periapsis to Neptune: 37162.242 km Semi-major axis: 544974.3798 km Jacobi Constant: 2.2153	Close passages to Triton during different phases along its orbit
<b>4:3 Resonant Orbit</b> 	Period: 16.83 days Periapsis to Triton: 1936.1990 km Periapsis to Neptune: 224819.7209 km Semi-major axis: 587722.0187 km Jacobi Constant: 2.9768	Closer passages to Triton during different phases along its orbit

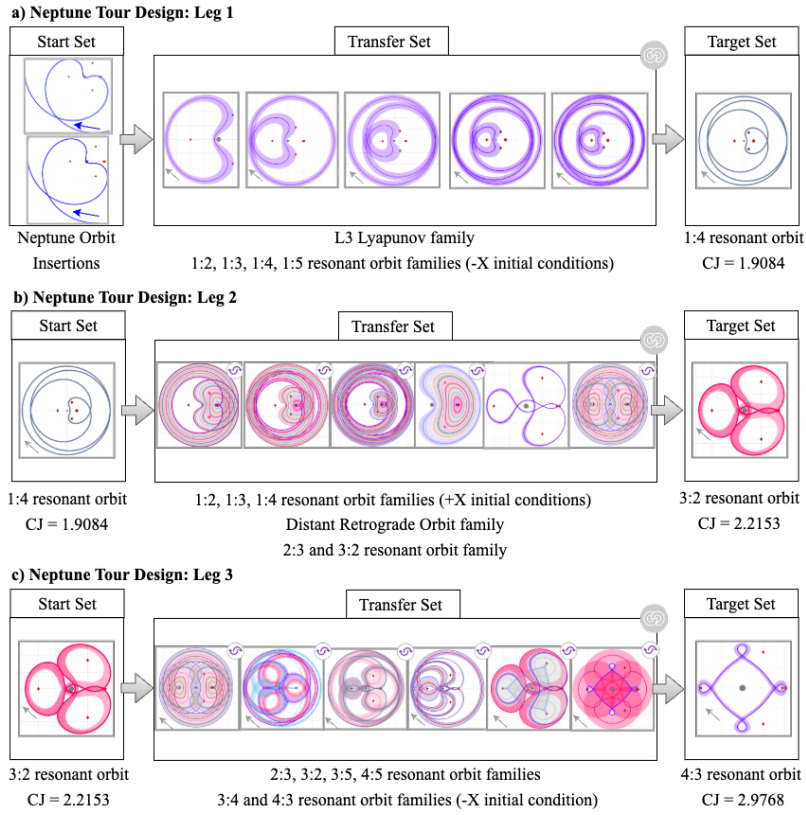
**Figure 6. Target orbits for the tour, characteristics, and scientific relevance.**

supplies sufficiently close passes of Triton to support investigating the presence of water oceans below the moon's surface.<sup>43</sup> This orbit is selected as the third target of the tours as it possesses the highest Jacobi constant of the three options.

### Tours Generation

The initial condition and target orbits define boundary conditions in three motion primitive graphs. In this paper, the first graph is used to design transfers between either of the two initial conditions and a 1 : 4 resonant orbit, as displayed in Figure 7a). The primitives connecting these boundary trajectories, included in the transfer set in Figure 7a), are selected based on similarity in geometry and Jacobi constant values to the trajectories of the initial and final conditions. Therefore, the primitives for this transfer are obtained from the  $L_3$  Lyapunov, 1 : 2, 1 : 3, 1 : 4, and 1 : 5 resonant orbit families with periapsis located on the  $-x$  axis. Loose constraints on the path and maneuver cost are imposed on the graph: the periapsis altitude with respect to Triton is constrained to possess values above 10 km and a maximum single  $\Delta v$  cannot exceed 5 km/s. From this graph, several motion primitive subsequences with different times of flight and geometries are obtained through the search algorithm described in Step 3 of the technical approach section. The initial guesses are then refined to minimize discontinuities and corrected. Three, selected initial guesses for this first leg of the tours are displayed in Figure 8a). These initial guesses possess distinct geometries and are constructed using distinct sequences of primitives.

The second leg of the tours is obtained by connecting the 1 : 4 resonant orbit to the 3 : 2 resonant orbit with a  $\sim 5000$  km flyby altitude with Triton. A set of primitives with specific geometries and



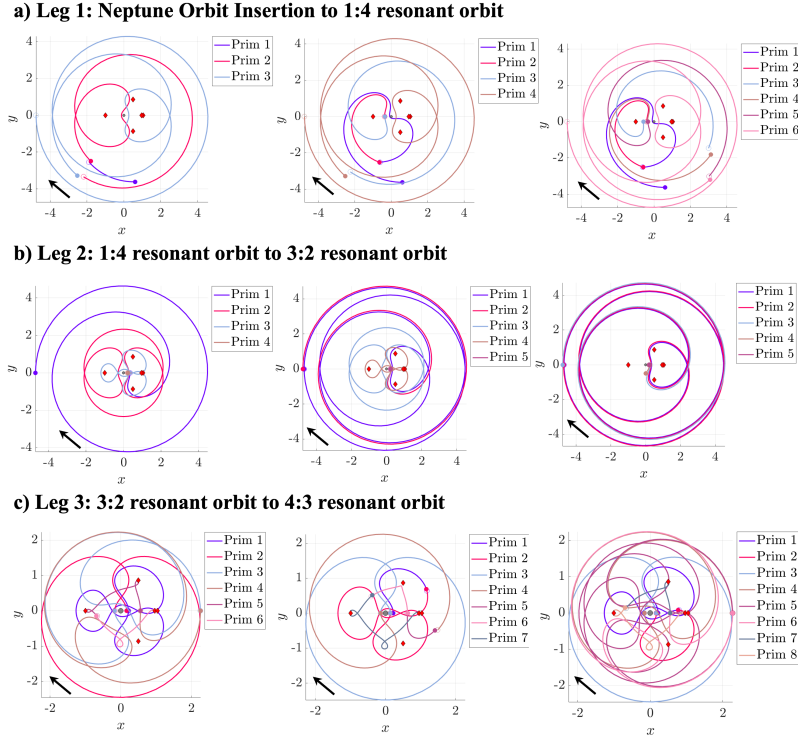
**Figure 7. High-level motion primitive graphs for transfers: a) from two initial conditions to a 1:4 resonant orbit; b) from a 1:4 resonant orbit to a 3:2 resonant orbit; and c) from a 3:2 resonant orbit to a 4:3 resonant orbit. The gray and purple round arrows at the top of a set or a primitive family represent full internal connectivity between the primitives of the relative set.**

Jacobi constant ranges are selected to form the transfer set shown in Figure 7b). In particular, the primitives employed in this graph include the 1 : 2, 1 : 3, and 1 : 4 resonant orbit families with periapsis located on the  $+x$  axis, the Distant Retrograde Orbit (DRO) family and some members of the 2 : 3 and 3 : 2 resonant orbit families. Then, the same constraints used for the first leg are applied to this second leg. The refined initial guesses obtained for the second leg of the tours are shown in Figure 8b). These transfers leave the outer region of the Neptunian system to bring the spacecraft much closer to the two primary bodies, supporting the analysis of Neptune and Triton.

Finally, a third leg is designed to connect the 3 : 2 resonant orbit with a 4 : 3 resonant orbit with a 300 km periapsis altitude with respect to Triton. For this transfer, primitives from the 2 : 3, 3 : 2, 3 : 4, 3 : 5, and some 4 : 3 resonant orbit families are included in the transfer set, as displayed in Figure 7c). After imposing the same constraints on altitude and maneuver cost and computing the motion primitive subgraph, a variety of initial guesses are generated and refined. A subset of three initial guesses is displayed in Figure 8c). These initial guesses are constructed using distinct primitive sequences and, therefore, possess distinct geometries.

One initial guess for each leg is used to generate tours as shown in Figure 9 a). First, an initial guess from each leg of the tours is independently corrected and optimized in the NT-CR3BP using collocation, as described in Step 5 of the technical approach. For the collocation mesh, each segment



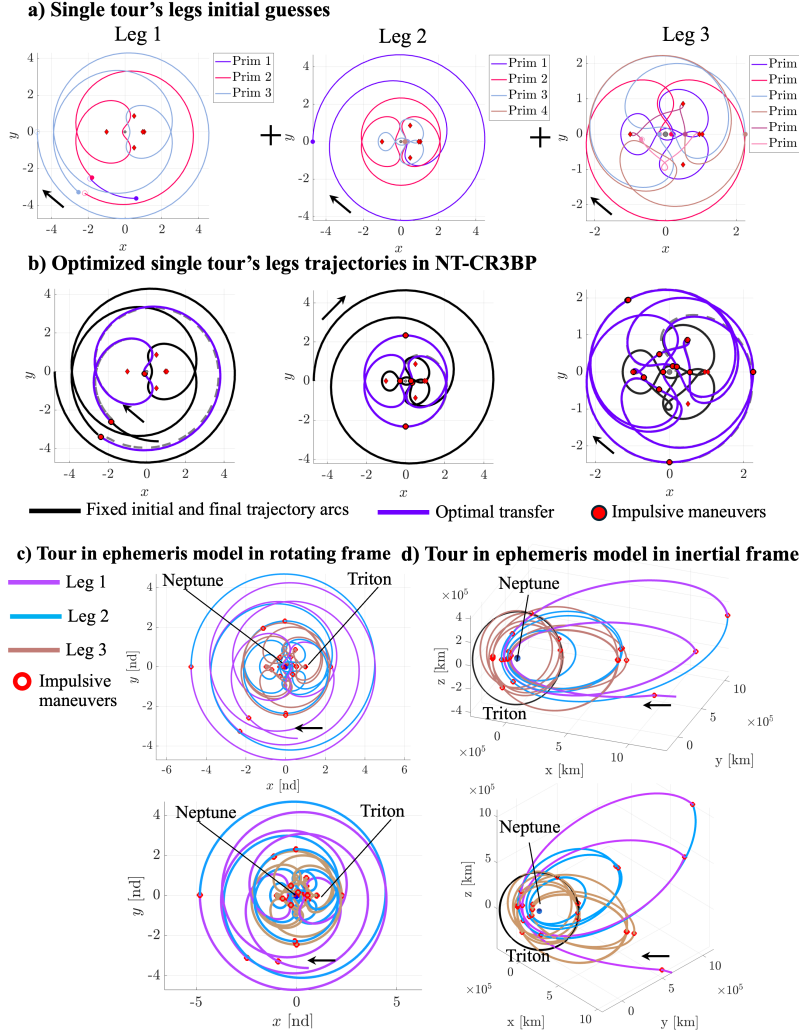


**Figure 8. Example of initial guesses for transfers: a) from one of two initial conditions to a 1:4 resonant orbit; b) from a 1:4 resonant orbit to a 3:2 resonant orbit; and c) from a 3:2 resonant orbit to a 4:3 resonant orbit.**

begins and ends at an apsis with respect to Neptune, then arcs are equally spaced in arclength within each segment. The number of arcs equally spaced in arclength depends on the primitives that compose the initial guess. Primitives with a long integration time are discretized using 8 – 14 arcs equally spaced in arclength, whereas shorter primitives are discretized with as few as 4 arcs with equal arclength within each segment. Each arc is then discretized using 7 collocation nodes. During correction and optimization, the spacecraft is constrained to maintain a minimum altitude of 10 km from each of Neptune and Triton to avoid obtaining trajectories that impact the primaries' surface. Finally, impulsive maneuvers are placed at the apsis with respect to Neptune and between different primitives. An example of the corrected and optimized trajectories for each initial guess is displayed in 9 b).

Finally, each leg is corrected in an ephemeris model. For the first leg, the initial epoch is selected as 3.5 days before the initial epoch of the NOI condition. Then, the final epoch of each leg supplies the initial epoch of the next leg. At this stage, additional revolutions along the target orbit of each leg are added to support performing scientific measurements or observations. The corrected paths within each leg are then patched together to form the full tour. An example of two complete tour trajectories is visualized in each of the rotating and inertial frames as displayed in Figures c) and d), respectively. For the first tour, the total flight time is  $TOF = 102.1562$  days while for the second  $TOF = 118.3814$  days, including 1 revolution along the 1 : 4 resonant orbit and 2 revolutions along each of the 3 : 2 and the 4 : 3 resonant orbits. Finally, the total maneuver requirement is  $\Delta v = 5.7168$  km/s for the first trajectory and  $\Delta v = 7.3423$  km/s, with maneuvers at the apses with respect to Neptune and between different primitives. The cost for these tours is higher than the total

cost for tours in the Uranian and Neptunian systems presented in previous studies.<sup>7,44</sup> However, differently from the existing designs, the target orbits selected for these tours are at a lower energy with respect to the initial conditions, leading to an increased  $\Delta v$  requirement. Additionally, gravity assists are traditionally used to reduce fuel consumption when reducing or increasing the energy during a planetary tour. In this work, we only leverage impulsive maneuvers to this scope. The placement of these maneuvers is also fundamental to reducing the final maneuver cost. Alternative maneuver placement strategies could be tested to reduce the maneuver requirements.



**Figure 9.** Example of Neptunian system tours obtained using shorter motion primitive subsequences.

## CONCLUSIONS

In this paper, scientific tours of the Neptunian system are constructed using a motion primitive trajectory design approach. In this version, the motion primitive graph is updated to support better estimating the potential sequential composability of primitives as well as improving the quality and diversity of primitive sequences identified from the graph. This paper also presents an improved

custom  $k$ -best path search algorithm to rapidly obtain diverse initial guesses from the motion primitive graph. This updated motion primitive trajectory design approach is used to design trajectories targeting a sequence of science orbits in the Neptunian system. Motion primitive graphs are constructed for each leg of the tours to connect sequential pairs of initial conditions and/or science orbits, to reduce the complexity of the problem. Geometrically distinct initial guesses for each leg are generated by searching each graph. These initial guesses are corrected and sequentially composed to design tours to study the Neptunian system.

## ACKNOWLEDGMENTS

Travel to this conference was supported by funding from the Strategic University Research Partnership program at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). The authors also thank Dr. Reza Karimi for providing the initial conditions associated with Neptune Orbit Insertion.

## REFERENCES

- [1] E. C. Stone and E. D. Miner, "The Voyager 2 Encounter with the Neptunian System," *Science*, Vol. 246, No. 4936, 1989, pp. 1417–1421.
- [2] National Academies of Sciences, Engineering, and Medicine, "NASA 2023 Decadal Survey," technical report, National Academies Press, Washington, D.C., 2023.
- [3] R. Helled, N. Nettelmann, and T. Guillot, "Uranus and Neptune: Origin, Evolution and Internal Structure," *Space Science Reviews*, Vol. 216, No. 3, 2020, p. 38.
- [4] National Aeronautics and Space Administration (NASA), "Neptunian Satellite Fact Sheet," 2023.
- [5] S. S. Shepard, "New moons of Uranus and Neptune," <https://sites.google.com/carnegiescience.edu/sheppard/home/newuranusneptunemoons>, 2024.
- [6] W. B. McKinnon and R. L. Kirk, "Chapter 40 - Triton," *Encyclopedia of the Solar System (Third Edition)* (T. Spohn, D. Breuer, and T. V. Johnson, eds.), pp. 861–881, Boston: Elsevier, third edition ed., 2014.
- [7] S. Campagnola, A. Boutonnet, W. Martens, and A. Masters, "Mission design for the exploration of Neptune and Triton," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 30, No. 7, 2015, pp. 6–17.
- [8] J. Melman, G. Orlando, E. Safipour, E. Mooij, and R. Noomen, "Trajectory Optimization for a Mission to Neptune and Triton," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, 18-21 August 2008.
- [9] G. E. Miceli, N. Bosanac, J. R. Stuart, and F. Alibay, "Motion Primitive Approach to Spacecraft Trajectory Design in the Neptune-Triton System," *AIAA SciTech Forum*, Orlando, FL, January 2024.
- [10] G. E. Miceli, N. Bosanac, and R. Karimi, "Generating the Trajectory Design Space for Neptunian System Exploration," *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, CO, August 2024.
- [11] T. R. Smith and N. Bosanac, "Constructing Motion Primitive Sets to Summarize Periodic Orbit Families and Hyperbolic Invariant Manifolds in a Multi-Body System," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, No. 1, 2022, p. 7.
- [12] 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.
- [13] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. London: Academic Press, 1967.
- [14] C. D. Murray and S. F. Dermott, *Solar system dynamics*. Cambridge university press, 1999.
- [15] C. H. Acton, "Ancillary data services of NASA's Navigation and Ancillary Information Facility," *Planetary and Space Science*, Vol. 44, No. 1, 1996, pp. 65–70. Planetary data system.
- [16] C. Acton, N. Bachman, B. Semenov, and E. Wright, "A look towards the future in the handling of space science mission geometry," *Planetary and Space Science*, Vol. 150, 2018, pp. 9–12. Enabling Open and Interoperable Access to Planetary Science and Heliophysics Databases and Tools.
- [17] R. S. Park, W. M. Folkner, J. G. Williams, and D. H. Boggs, "The JPL Planetary and Lunar Ephemerides DE440 and DE441," *The Astronomical Journal*, Vol. 161, feb 2021, p. 105.
- [18] D. J. Grebow and T. A. Pavlak, "MCOLL: Monte Collocation Trajectory Design Tool," *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson, Washington, August 20-24, 2017, 2017.

- [19] B. A. Conway, *Spacecraft Trajectory Optimization*. New York: Cambridge University Press, 2010.
- [20] P. Williams, "Hermite-Legendre-Gauss-Lobatto Direct Transcription in Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 2009, pp. 1392–1395.
- [21] R. E. Pritchett, *Strategies for Low-Thrust Transfer Design Based on Direct Collocation Techniques*. Thesis, Purdue University Graduate School, 2020.
- [22] C. De Boor, "Good Approximation by Splines with Variable Knots. II," *Proceedings of the Conference on the Numerical Solution of Differential Equations*, Dundee, Scotland, 1973.
- [23] R. Russell and J. Christiansen, "Adaptive Mesh Selection Strategies for Solving Boundary Value Problems," *SIAM Journal on Numerical Analysis*, Vol. 15, No. 1, 1978, pp. 59–80.
- [24] MATLAB, *version R2006a*. Natick, Massachusetts: The MathWorks Inc., 2006–2009.
- [25] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
- [26] R. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," *Advances in Knowledge Discovery and Data Mining* (J. Pei, V. Tseng, L. Cao, H. Motoda, and G. Xu, eds.), Springer Berlin, Heidelberg, 2013, p. 160–172.
- [27] A. Wolek and C. A. Woolsey, *Model-Based Path Planning*, pp. 183–206. Cham: Springer International Publishing, 2017.
- [28] N. Bosanac and M. Joyner, "Data-Driven Summary of Continuous Thrust Trajectories in a Low-Fidelity Model of Cislunar Space," *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, CO, August 2024.
- [29] N. Bosanac, "Data-Mining Approach to Poincaré Maps in Multi-Body Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 6, 2020, pp. 1190–1200.
- [30] S. Bonasera and N. Bosanac, "Applying data mining techniques to higher-dimensional Poincaré maps in the circular restricted three-body problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 133, Dec. 2021, p. 51.
- [31] N. Bosanac, "Data-Driven Summary of Natural Spacecraft Trajectories in the Earth-Moon System," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, August 2023.
- [32] G. E. Miceli, N. Bosanac, M. A. Mesarch, D. C. Folta, and R. L. Mesarch, "Clustering Approach To Identifying Low Lunar Frozen Orbits In A High-Fidelity Model," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, August 2023.
- [33] R. L. Spear and N. Bosanac, "Data-Driven Categorization of Spacecraft Motion with Uncertainty in the Earth-Moon System," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, August 2023.
- [34] O. Ore, *Graphs and their uses*, Vol. 34. Cambridge University Press, 1990.
- [35] E. Frazzoli, *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.
- [36] A. Majumdar and R. Tedrake, "Funnel Libraries for Real-Time Robust Feedback Motion Planning," *Int. J. Robot. Res.*, Vol. 36, No. 8, 2017, pp. 947–982. DOI: <https://doi.org/10.1177/0278364917712421>.
- [37] D. J. Grymin, C. B. Neas, and M. Farhood, "A hierarchical approach for primitive-based motion planning and control of autonomous vehicles," *Robotics and Autonomous Systems*, Vol. 62, No. 2, 2014, pp. 214–228.
- [38] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107.
- [39] J. Yen, "An Algorithm for Finding Shortest Routes from All Source Nodes to a Given Destination in General Networks," *Quarterly of Applied Mathematics*, Vol. 27, 1970, pp. 526–530.
- [40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pp. 287–290, 2022.
- [41] T. Chondrogiannis, P. Bouros, J. Gamper, U. Leser, and D. B. Blumenthal, "Finding k-shortest paths with limited overlap," *The VLDB Journal*, Vol. 29, No. 5, 2020, pp. 1023–1047.
- [42] C. Dumas, B. A. Smith, and R. J. Terrile, "Hubble Space Telescope NICMOS Multiband Photometry of Proteus and Puck," *The Astronomical Journal*, Vol. 126, aug 2003, p. 1080.
- [43] C. J. Cochrane, R. R. Persinger, S. D. Vance, E. L. Midkiff, J. Castillo-Rogez, A. Luspay-Kuti, L. Liuzzo, C. Paty, K. L. Mitchell, and L. M. Prockter, "Single- and Multi-Pass Magnetometric Subsurface Ocean Detection and Characterization in Icy Worlds Using Principal Component Analysis (PCA): Application to Triton," *Earth and Space Science*, Vol. 9, No. 2, 2022.
- [44] A. Simon, F. Nimmo, and R. C. Anderson, "Journey to an Ice Giant System: Uranus Orbiter and Probe," Planetary Mission Concept for the 2023–2032 Planetary Science Decadal Survey, June 7 2021. Retrieved May 1, 2022.