Data-Driven Spacecraft Trajectory Design For Planetary System Exploration

by

Giuliana Elena Miceli

B.S., Università degli Studi di Padova, 2018

M.S., Institut Supérieur de l'Aéronautique et de l'Espace, 2020

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Ann and H.J. Smead Department of Aerospace Engineering Sciences

2025

Committee Members:

Prof. Natasha Bosanac, Chair

Prof. Daniel Scheeres

Prof. Marcus Holzinger

Dr. Jeffrey Stuart

Dr. Reza Karimi

Miceli, Giuliana Elena (Ph.D., Aerospace Engineering Sciences)

Data-Driven Spacecraft Trajectory Design For Planetary System Exploration

Thesis directed by Prof. Natasha Bosanac

The exploration of multi-body planetary systems is a crucial task for the enhancement of science and the establishment of a human presence in space. However, trajectory design in these multi-body systems remains a challenging endeavor due to the complex and often chaotic nature of the underlying dynamics. Traditional approaches are usually insufficient as they might require a priori knowledge of the system's dynamics, or heavily rely on a human-in-the-loop. In designing trajectories for a planetary mission, another challenge arises from the selection of long-term target orbits about a planetary moon with limited station-keeping requirements.

This dissertation presents a data-driven approach to designing trajectories in a multi-body system and identifying low-altitude, long-term stable orbits around a moon. First, clustering is used to summarize natural dynamical structures in the Circular Restricted Three-Body Problem (CR3BP) and isolate trajectories with a similar perilune sequence in a high-fidelity model. In the first case, representatives of the clusters become motion primitives: building blocks of motion inspired by robotic path planning. Motion primitives are used to create a graph that describes the primitives' composability. The graph is searched to obtain the best diverse primitive sequences, which are then refined to obtain trajectory initial guesses. Finally, these paths are corrected and optimized in an ephemeris model to provide a variety of solutions to move across different regions of a planetary system. This approach is demonstrated by computing a diverse set of transfers in the Neptune-Triton system. In the second case, representatives of trajectories are identified as the most bounded members of the group. The representatives are further clustered by geometry to identify families of candidate frozen and quasi-frozen orbits near the Moon, existing across a wide combination of orbital elements.

Dedication

To Daniela and Erasmo, my safe harbor.

To Flora and Delia, my guiding light.

To Jacopo, the wind in my sail.

Acknowledgements

I would first like to thank my advisor, Dr. Natasha Bosanac, for her advice, her continuous support, and the countless hours spent discussing research ideas and providing feedback. Thank you for being a brilliant professor, a great speaker, and an amazing researcher. Through your guidance, I learned to be somewhat better in all these areas, but your example will keep inspiring me. Then, I would like to acknowledge my research mentors: Dave Folta, Michael and Becky Mesarch, Jeff Stuart, Reza Karimi, and Farah Alibay. Through different projects, you have all welcomed me into the team as a peer and helped me navigate the most challenging research problems. Your kindness, your encouragement, and your professional support helped me become a better astrodynamicist and a better person. I would also like to express my gratitude for the partnership with NASA Goddard Space Flight Center (NASA Grant 80NSSC22K1151) and the Strategic University Research Partnership program at Jet Propulsion Laboratory (NASA Grant 80NM0018D0004) for supporting this research. Next, I would like to recognize my lab mates, old and new, for being the best companions along this PhD journey. Thanks to Stefano, Jack, Ian, Thomas and Kristen for showing me the look of successful PhD students; thanks to Cole, Miguel, Max, Austin and Sai for sharing joys and sorrows of research inside and outside the lab; finally, thanks to Renee for being the best friend to embark on this adventure. Last but not least, I would like to give a special thank to my parents, Daniela and Erasmo, my sisters, Flora and Delia, and my partner, Jacopo, for the immense role each of them played in helping me succeed in this PhD, but also in becoming the person I am today. There are not enough words to express the love, gratitude, and respect I have for each one of you. Thank you for always being by my side in this lifetime.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

Missions to explore Neptune offer a unique opportunity to deepen our understanding of the outer Solar System. They promise to reveal critical insights into the planet's formation, internal structure, atmospheric composition, and magnetic environment, as well as to uncover new information about its complex and diverse moon system [81]. A critical task for developing these missions is designing trajectories for a spacecraft to support scientific observations. However, this task might become quite challenging when considering the conditions characterizing a mission to an ice giant. Among these, the high-energy state that follows the interplanetary transfer from Earth imposes the main constraint on the trajectory design because a spacecraft approaching Neptune from deep space might be traveling at a very high v_∞. The large maneuver required for gravitational capture significantly limits the spacecraft's maneuvering capability during the mission's scientific phases. Secondly, the high heliocentric distance of Neptune constrains the maneuvering locations to specific states along the trajectory due to the limited visibility from Earth. Maneuvers are indeed very sensitive locations during a trajectory, primarily when they are performed in the proximity of a planet; therefore, they must be tracked from Earth to assess the possible errors and compute corrections in a timely manner. Such a constraint limits the trajectory design trade space and consequently the science return of the mission. Finally, the chaos of a multi-body gravitational system composed of Neptune and its moons, along with the constraints derived from hardware parameters or operational requirements, adds further complexity to the trajectory design objective.

One common approach to designing science trajectories in a planetary system involves using patched conics. The patched conics technique approximates the gravitational environment of a multi-body system as a sequence of two-body problems [106] to design trajectories around different objects. This approach has been employed in mission concept studies presented by Marley et al. [66] and Masters [67], where multiple Triton flybys are utilized to investigate the Neptunian moon. Designing multi-body system trajectories using a patched conics approach may limit the exploration of the solution space and result in suboptimal combinations of propellant mass usage and flight. To better capture the influence of multiple gravitational bodies, more advanced techniques rely on simplified multi-body models such as the circular restricted three-body problem (CR3BP), which considers the gravitational interaction between a planet, a moon, and a massless spacecraft. For instance, Melman et al. computed a transfer for a spacecraft from a Neptune-centered orbit to a circular near-polar orbit around Triton in the CR3BP [71].

When considering trajectory design for planetary exploration, another challenge that arises is the selection of meaningful, long-term, stable orbits about a celestial body. Such orbits should enable scientific measurements while minimizing the need for orbit maintenance over an extended period. For this purpose, frozen orbits–or even quasi-frozen orbits with a bounded variation in the orbital elements–have been of significant interest for designing mission orbits that require little maintenance over long time intervals. These trajectories can support scientific missions, the placement of critical infrastructure, and extended imaging of the celestial body's surface. Around the Moon, previous missions have already leveraged frozen and quasi-frozen lunar orbits identified in high-fidelity gravity models to limit station-keeping maneuver requirements, including Lunar Prospector (1997-1999) and the Lunar Reconnaissance Orbiter (2009 - present)[36, 5]. Researchers have identified lunar frozen orbits using analytical approximations in truncated dynamical models or numerically in high-fidelity dynamical models. The first approach provides an exact mathematical formulation of the initial conditions but can consider only a simplified model of the Moon's gravity field. On the other hand, a numerical approach in a high-fidelity model increases accuracy but limits the generalization of the initial conditions of such orbits to the analyzed dataset, potentially increasing computational time. More generally, identifying frozen orbits is not a trivial task, especially for bodies with complex gravity fields.

1.2 Previous Contributions

1.2.1 Planetary Exploration in Multi-Body Systems

Mission concepts to visit the ice giants have been explored since 1992, [102], but the interest in these bodies has been recently renewed by the planetary science community [77, 33]. Due to this interest and the challenge that such a mission poses, many authors have been investigating various new concepts for Neptune missions. In 2014, Masters et al. [67] published a high-level scientific mission analysis to explore Neptune and Triton. In 2021, Rymer et al. [97] detailed the concept for the Neptune Odyssey mission: a Flagship-class orbiter and atmospheric probe to the Neptune–Triton system. Finally, Turrini et al. [104] proposed the ODINUS mission concept, which involves sending two twin spacecraft to be placed around each of the ice giants to conduct a simultaneous comparative study of the planets. Across these studies, the trajectory design problem in the Neptunian system has been addressed through the use of patched conics, such as in mission concept studies presented by Marley et al. [66], where petal orbits are used to obtain subsequent flybys of Triton. This technique has often been employed in the exploration of planetary systems with multiple sizable moons, particularly in the design of moon tours. For example, the Galileo mission [24] employed a set of petal orbits to perform flybys of the Jovian moons.

Other authors employed multi-body trajectory design for a planetary exploration purpose, proving that a higher fidelity representation of the dynamics of n-body planetary systems provides trajectories that exploit the natural dynamics. An example is provided by Campagnola et al. [16], who demonstrated that the use of multi-body dynamics improves the maneuver cost of a moon tour in the Jovian system when compared to previous designs for a Europa mission that employed patched conics and full-ephemeris models. In addition to studies, past and present missions have proven the feasibility of flying trajectories designed in multi-body dynamics. This is the case of the James Webb Space Telescope, which reached a halo orbit around the Sun-Earth L2 point, and CAPSTONE, which arrived at a near rectilinear halo orbit in the Earth-Moon system [43, 19].

1.2.2 State of the Art in Trajectory Design

Trajectory design in multi-body systems is often a complex task due to its chaotic and nonlinear dynamics. While dynamical models exist to simplify this process, such as the circular restricted three-body problem, the interpretation of the solution space can still be nontrivial [45]. In these instances, selecting trajectories that meet the mission requirements often involves techniques that simplify the solution space without modifying its properties. One of such techniques is Poincaré mapping, which is an efficient tool from dynamical systems theory, often employed in trajectory design.

A Poincaré map represents the states where a dynamical flow intersects a surface of section defined to be transverse to the flow. In the application of the Poincaré map to astrodynamics, the surface of section is often represented as a hyperplane at a fixed value of the position space; meanwhile, the flow is defined by the nonlinear equations of motion describing the n-body dynamics in space [4]. The map realized by plotting the crossings represents a lower-dimensional visualization of the dynamics. The reduced dimensions of the new solution space do not modify the dynamical properties of the flow, but rather reveal patterns in the motion and aid in identifying segments for initial trajectory guesses. The mapping facilitates the identification of transfer arcs that leverage the system's natural dynamical structures to generate trajectory initial guesses.

This technique has been proven helpful in a variety of applications in the CR3BP, particularly in the planar problem at a specific energy level, due to the map's ability to reduce the initial guess search to a two-dimensional problem [45]. However, as the trajectory design problem becomes more complex, for example, in cases involving spatial trajectories and multiple energy levels, the use of a Poincaré map becomes challenging and requires the introduction of specific visualization techniques [45]. Moreover, constructing a surface of section for the map is a manual process that requires particular expertise and insight into the design problem and does not allow the visualization of the initial guess geometry. For this reason, traditional techniques for trajectory design in multi-body dynamics are limiting in the exploration of the solution space and usually require a human-in-the-loop.

Once the initial guesses are obtained, they are corrected to achieve a continuous trajectory and optimized to minimize the total maneuver cost or the time of flight. Correction methods such as multiple shooting and collocations have been widely used to correct trajectories in multi-body dynamics [23]. However, the success of these methods depends on the quality of the initial guess and the characteristics of the numerical scheme used, which makes robust and efficient design strategies essential for supporting complex mission architectures and real-time operations. In trajectory optimization, a variety of tools have been used to explore multi-body trajectory design trade spaces. This is the case of the DESTINY+ mission, where multiobjective evolutionary computation is used to efficiently obtain the approximate Pareto front [110]. Similarly, the Johnson Space Center's (JSC) Copernicus spacecraft trajectory optimization tool was employed in constructing the trade space for the Orion Exploration Mission 1 [25]. However, such tools can be computationally expensive, especially for a large trade space, relying on parallelization and approximation techniques [25, 110].

1.2.3 State of the Art in Frozen Orbit Analysis

Since the early 1960s, astrodynamicists have studied frozen orbits: trajectories that exhibit small variations in the orbital elements relative to a selected celestial body over long time intervals [57]. Due to their properties, frozen orbits have been studied for several bodies, including planets like the Earth and Mars, planetary satellites like the Moon, and small bodies such as asteroids [22, 99, 64, 37]. In particular, around the Moon, several authors have investigated analytical methods to identify such orbits. As an example, Ely examined the evolution of the orbital elements in a point mass lunar gravity model with a third-body perturbation from the Earth to identify elliptical and inclined lunar frozen orbits [30]. Folta and Quinn employed a similar approach to identify lunar frozen orbits, which are then numerically simulated in a higher-fidelity ephemeris model and leveraged for maneuver design [37]. Additionally, Elipe and Lara used corrections and continuation algorithms to compute frozen orbits in a lunar gravity model that captures the first seven zonal harmonic terms, identifying three distinct families of lunar frozen orbits across various eccentricities and inclinations [29]. Lara, Ferrer, and De Saedeleer then used an averaged Hamiltonian formulation of a lunar gravity model with the first 50 zonal harmonics and the point mass gravity of the Earth to examine the long-term behavior of low lunar polar orbits [61]. A wide variety of researchers have used similar approaches to identify frozen orbits in low-order spherical harmonic gravity models of the Moon that are augmented by the gravitational influence of the Earth [98, 1, 105].

Researchers have also demonstrated that numerical methods can be used to determine the initial conditions for frozen orbits, as they provide a more accurate representation of the Moon's dynamical field. For example, Russell and Lara identified families of multi-revolution periodic orbits near the Moon via numerical integration and differential corrections in an Earth-Moon restricted three-body problem, which is augmented with a 50×50 lunar gravity model [96]. In general, this approach requires a large number of trajectories to be integrated over some period of time to analyze the effect of the Moon's irregular gravitational field on the orbital elements. However, the analysis of a large dataset of numerical data is often not trivial without a priori knowledge of the data corresponding to frozen and quasi-frozen orbits.

1.3 Dissertation Overview

The necessity for a dedicated mission to an ice giant like Neptune requires the ability to quickly and efficiently explore the trajectory design solution space in a multi-body dynamics environment. A mission to planetary systems would also benefit from an extended scientific phase targeting a moon, which would require the selection of a long-term bounded orbit with low maneuver requirements. **This dissertation focuses on data-driven approaches to designing constrained spacecraft trajectories within multi-body planetary systems and identifying long-term stable orbits around a moon.**

First, clustering is used to group a set of high-fidelity trajectories via the evolution of their perilunes. The clusters are used to identify potential lunar frozen orbits at different inclinations around the Moon. In the second part of this dissertation, clustering is also leveraged to identify a summary of the fundamental solutions of the Neptune-Triton CR3BP. This summary is used to extract fundamental blocks of motion, or motion primitives, consisting of geometrically distinct arcs that govern the motion in the system. The motion primitives are leveraged in a multi-body trajectory design approach, presenting an updated version of the work originally developed by Smith and Bosanac [101]. In this work, selected motion primitives are used to inform the creation of a graph, representing the connectivity between states along the primitives. The graph is searched through a custom search algorithm to identify unique sequences of primitives that minimize the overall discontinuity in velocity direction among the primitives. These sequences are used to generate a second graph that reflects the connectivity of the selected primitive arcs in position space. Then, a trajectory initial guess is obtained by searching the path that minimizes the position discontinuity between the primitives. Finally, the initial guesses are corrected and optimized in the restricted problem and in an ephemeris model.

1.3.1 Organization

This dissertation is composed of eight chapters, organized as follows:

Chapter 2: This chapter presents the dynamical models employed in this work. The dynamics of a spacecraft subject to the influence of a planet and its main moon is modeled using the circular restricted three-body problem, and then corrected using a point-mass ephemeris model. The first sections of this chapter are dedicated to an in-depth discussion about the CR3BP, its formulation, and its fundamental solutions. The last section provides an overview of the formulation of the ephemeris model and the transformation between the different reference frames used in the different dynamical models.

Chapter 3: The data-driven approach described in this work leverages the use of clustering to create summaries of the type of motion governing the system. This chapter presents an overview of the clustering algorithms, namely the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) and the Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The first algorithm is employed in identifying the candidate lunar frozen orbits, while a combination of both algorithms is used to compute the motion primitives.

Chapter 4: This chapter presents a data-driven approach to autonomously identify long-term stable orbits from a set of periapsis trajectories. First, a wide array of high-fidelity lunar orbits is propagated for six months or until impact. The perilune evolution of each orbit is analyzed as the variation of e and ω in time. Then, the perilune evolutions in time are sampled and described with a feature vector capturing their shape and phasing in time. Clustering is applied to the feature vectors to identify perilunes with similar spatio-temporal characteristics. The most bounded members of each cluster are identified as the cluster representatives and then grouped by shape. From these groups, perilune with long-term bounded evolutions are isolated. The most bounded elements within the selected clusters become candidates for frozen and quasi-frozen orbits.

Chapter 5: This chapter applies the approach presented in Chapter 4 to identify candidates for lunar frozen orbits, starting from sets of high-fidelity lunar trajectories. Moreover, an analysis of the short-, medium-, and long-term stable orbits is presented, based on the evolution of their perilune trajectories in time and their initial conditions.

Chapter 6: Motion primitives have been used in robotics to solve a variety of path planning problems. This chapter presents a motion primitives approach to trajectory design in multi-body systems. First, motion primitives are extracted from the fundamental solutions of the Neptune-Triton CR3BP. These are summarized into a graph that describes their sequential composability in phase space. The graph is searched through a primitive-based search algorithm to obtain primitive sequences, which are refined to compose trajectory initial guesses. Continuous, impulsive trajectories are finally obtained from the refined primitive sequences via a collocation-based optimization algorithm. A trajectory design example is provided to support the description of the technical approach.

Chapter 7: The technical approach presented in Chapter 6 is applied to a mission for the scientific exploration of the Neptunian system. In this chapter, three trajectory design cases are presented and analyzed. First, a high-energy trajectory is designed from a Neptune orbit insertion condition to a science orbit. Then, a medium-energy transfer is designed between two distinct resonant orbits with low periapsis passages at Triton. Finally, a low-energy trajectory is designed to reach a low prograde orbit about Triton from a Neptune-centered orbit.

Chapter 8: This chapter presents a summary of the work and draws conclusions on the obtained results. Finally, a concluding section discusses the potential future development and applications of the proposed research.

1.4 Contributions

To support trajectory design for planetary system exploration, this dissertation presents a data-driven approach to constructing transfers and identifying target orbits in a multi-body system. The completion of this research resulted in the following contributions:

(1) Updates on the motion-primitive trajectory design approach by Smith and Bosanac [101] through:

  (a) Curvature-based sampling to extract geometrically different primitives via 1) clustering for hyperbolic invariant manifolds and 2) an analytical approach for periodic orbits

  (b) Definition of motion primitive-informed graph using samples along motion primitives as nodes

  (c) Development of a graph search algorithm to rapidly identify diverse sequences of motion primitives

  (d) Development of graph-based refinement processes to rapidly produce geometrically diverse and smoother initial guesses from motion primitive sequences

  (e) Inclusion of constraints on distance from primaries and maneuver magnitude on the motion primitive graph

   (f) Correction and optimization of trajectories in an ephemeris model with maneuvers and position constraints

(2) Application of the motion primitive approach to design constrained trajectories supporting scientific goals in the Neptunian system

(3) Using clustering to summarize low-altitude orbits around celestial bodies with irregular gravity fields by:

   (a) Development of a clustering approach to summarize the solution space around a celestial body in a high-fidelity model

   (b) Identification and characterization of long-term stable bounded orbits with different geometries

   (c) Characterization of initial conditions leading to short-term stable and unbounded orbits

Chapter 2

Dynamical Models

2.1 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, referred to as the NT-CR3BP throughout this dissertation. In the NT-CR3BP, Neptune is the first primary body B_1, Triton is the second primary body B_2, and the spacecraft is the third body B_3. B_1 and B_2 are modeled as point masses, moving in circular orbits about their barycenter, and the spacecraft is assumed to possess a negligible mass with respect to the two primaries. Given the eccentricity of Triton's orbit around Neptune, $e_T = 0.000016$, and its mass, which is two orders of magnitude bigger than the other moons in the system [83], these assumptions are reasonable.

2.1.1 Equations of Motion

Newtonian mechanics is used to derive the equations of motion of the system described in Figure 2.1. In this conceptual representation, Neptune, Triton, and the spacecraft are represented in a configuration space with respect to two reference frames: an inertial reference frame $\mathcal{I}$ with axes $\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}}$ and a rotating reference frame $\mathcal{R}$ with axes $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$, both centered on the system barycenter. In the inertial frame, the positions of the primaries B_1, B_2 and the spacecraft B_3 with respect to the origin are, respectively

$$\begin{aligned}\tilde{\boldsymbol{R}}_1 &= \tilde{X}_1\hat{\boldsymbol{X}} + \tilde{Y}_1\hat{\boldsymbol{Y}} + \tilde{Z}_1\hat{\boldsymbol{Z}} \\ \tilde{\boldsymbol{R}}_2 &= \tilde{X}_2\hat{\boldsymbol{X}} + \tilde{Y}_2\hat{\boldsymbol{Y}} + \tilde{Z}_2\hat{\boldsymbol{Z}} \\ \tilde{\boldsymbol{R}}_3 &= \tilde{X}_3\hat{\boldsymbol{X}} + \tilde{Y}_3\hat{\boldsymbol{Y}} + \tilde{Z}_3\hat{\boldsymbol{Z}}\end{aligned} \tag{2.1}$$

where the tilde indicates dimensional quantities. The position of the spacecraft with respect to any primary in the inertial frame is computed as

$$\tilde{\boldsymbol{R}}_{i,3} = (\tilde{X}_3 - \tilde{X}_i)\hat{\boldsymbol{X}} + (\tilde{Y}_3 - \tilde{Y}_i)\hat{\boldsymbol{Y}} + (\tilde{Z}_3 - \tilde{Z}_i)\hat{\boldsymbol{Z}} \tag{2.2}$$

where the subscript i indicates a primary, thus $i = [1, 2]$ [103].

In this system, the spacecraft moves under the influence of the gravitational force exerted by the primaries. From these forces, the scalar potential function acting on B_3 is computed as

$$\tilde{U}_3 = \frac{\tilde{G}\tilde{M}_1}{\tilde{R}_{1,3}} + \frac{\tilde{G}\tilde{M}_2}{\tilde{R}_{2,3}} \tag{2.3}$$

where $\tilde{G}$ is the gravitational constant, $\tilde{M}_1$ and $\tilde{M}_2$ are the mass of the two primaries and $\tilde{R}_{i,3}$ denotes the distance between the spacecraft and a primary. The gravitational potential is used to

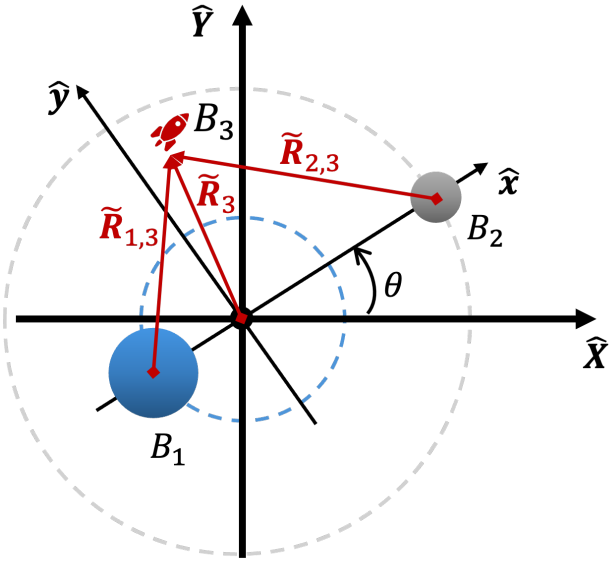


Figure 2.1: Configuration of B_1, B_2 and B_3 with respect to an inertial reference frame $\mathcal{I}$ with axis $\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}}$ and a rotating reference frame $\mathcal{R}$ with axis $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$

derive the acceleration that the gravitational force of the two primaries exerts on the spacecraft using Newton's second law of motion. In particular, a conservative force $\boldsymbol{F}$ such as gravity is derived from its potential function as the derivative of the potential with respect to an inertial observer [103]. Therefore, the acceleration on B_3 due to the gravitational pull of the two primaries is derived as

$$M_3\ddot{\tilde{\boldsymbol{R}}}_3 = \boldsymbol{F}_3 = \boldsymbol{\nabla}\tilde{U}_3 \tag{2.4}$$

where the notation $\ddot{(.)}$ indicates the second-time derivative with respect to an inertial observer. Here, the mass of the spacecraft is negligible with respect to the mass of the primaries, and it is simplified. Therefore, the acceleration on the spacecraft is equal to

$$\ddot{\tilde{\boldsymbol{R}}}_3 = \nabla\tilde{U}_3 = \frac{-\tilde{G}\tilde{M}_1}{\tilde{R}_{1,3}^3}\tilde{\boldsymbol{R}}_{1,3} - \frac{\tilde{G}\tilde{M}_2}{\tilde{R}_{2,3}^3}\tilde{\boldsymbol{R}}_{2,3} \tag{2.5}$$

The equation 2.5 can be rewritten as three scalar second-order ordinary differential equations (ODEs), which describe the motion of B_3 under the gravitational influence of B_1 and B_2. However, to solve the system of equations, it would be necessary to include the three scalar ODEs describing the motion of the first primary and the three scalar ODEs describing the motion of the second primary, creating a system of 9 second-order ODEs or 18 first-order ODEs. A solution to this system of equations would require 18 constants, but only 10 constants are obtained from the conservation of linear momentum, the angular momentum, and the energy in a closed system [103]. Due to the lack of a general solution for the three-body problem, the dynamics is reformulated with the assumptions described before as the CR3BP. Even though this dynamical model does not possess analytical solutions, it provides a simpler approximation of the dynamics of a three-body system, which is acceptable for applications, such as the Sun-Earth system or the Neptune-Triton system, as in this case [103].

Trajectories in this model are generated and analyzed using nondimensional coordinates in the Neptune-Triton rotating frame. Quantities are nondimensionalized using three characteristic quantities l^*, m^*, and t^* [103]. The characteristic quantities are computed from the properties of the system: l^* is the constant distance between B_1 and B_2, m^* is equal to the sum of the primaries' masses, and $t^* = 1/\tilde{n}$, where $\tilde{n}$ is the mean motion of the primary system, defined as

$$\tilde{n} = \left(\frac{\tilde{G}m^*}{l^{*3}}\right)^{1/2} \tag{2.6}$$

The characteristic quantities for the NT-CR3BP are derived from the system's physical parameters [82, 83] and are equal to $m^* \approx 1.024569 \times 10^{26}$ kg, $l^* = 354,760$ km, and $t^* \approx 8.081353 \times 10^4$ s. From m^*, the mass ratio of the system is defined as

$$\mu = \frac{\tilde{M}_2}{m^*} \tag{2.7}$$

where the mass ratio of the Neptune-Triton system is $\mu \approx 0.00020895$.

The rotating frame $\mathcal{R}$ is defined with the origin at the Neptune-Triton barycenter and axes $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$: $\hat{\boldsymbol{x}}$ is directed from B_1 to B_2, $\hat{\boldsymbol{z}}$ is in the direction of the orbital angular momentum vector of the primary system, and $\hat{\boldsymbol{y}}$ completes the right-handed triad [103]. A planar representation of $\mathcal{R}$ is shown in Figure 2.2. With respect to a general inertial frame where $\hat{\boldsymbol{z}} = \hat{\boldsymbol{Z}}$, the angle between the $\hat{\boldsymbol{x}}$ and the $\hat{\boldsymbol{X}}$ axis is defined at a nondimensional time t as $\theta = nt + \theta_0$, where $n = \tilde{n}t^*$ and θ_0 is the initial value of the rotation between the two frames, as shown in Figure 2.1. In this frame, the nondimensional state of the spacecraft is defined as $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$.

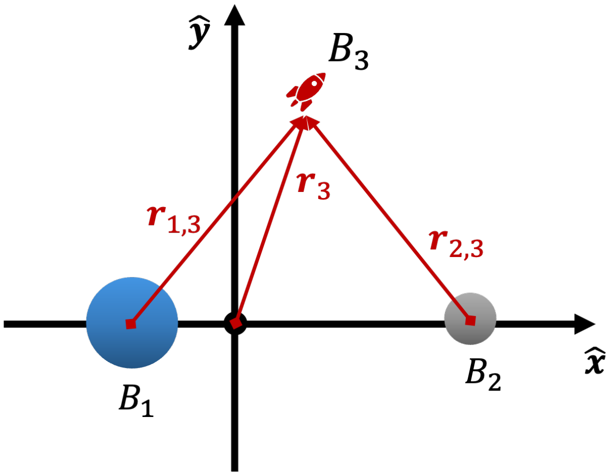


Figure 2.2: Configuration of B_1, B_2 and B_3 with respect to the rotating reference frame $\mathcal{R}$ with axes $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$

The characteristic quantities are used to nondimensionalize the equations describing the spacecraft acceleration in the inertial frame. Thus, Equation 2.5 is rewritten as

$$\ddot{\boldsymbol{R}}_3 = \frac{-(1-\mu)}{R_{1,3}^3}\boldsymbol{R}_{1,3}^3 - \frac{\mu}{R_{2,3}^3}\boldsymbol{R}_{2,3}^3 \tag{2.8}$$

With the use of the transport theorem, the acceleration of B_3 with respect to an observer in the rotating frame $\mathcal{R}$ and expressed independent of a coordinate frame is

$$\ddot{\boldsymbol{R}}_3 = -2(\boldsymbol{\omega}_{\mathcal{R},\mathcal{I}} \times \dot{\boldsymbol{R}}_3) - (\boldsymbol{\omega}_{\mathcal{R},\mathcal{I}} \times (\boldsymbol{\omega}_{\mathcal{R},\mathcal{I}} \times \dot{\boldsymbol{R}}_3)) - \frac{-(1-\mu)}{R_{1,3}^3}\boldsymbol{R}_{1,3}^3 - \frac{\mu}{R_{2,3}^3}\boldsymbol{R}_{2,3}^3 \tag{2.9}$$

where $\boldsymbol{\omega}_{\mathcal{R},\mathcal{I}}$ is the angular velocity of $\mathcal{R}$ with respect to $\mathcal{I}$.

Since the motion of a spacecraft in the CR3BP is most commonly evaluated in the rotating frame, the Equation 2.9 should be evaluated in $\mathcal{R}$. In this reference frame, the position of the spacecraft is defined as $\boldsymbol{r}_3 = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$, where the subscript in the position component is dropped for simplicity. Then, the positions of the primaries from the inertial frame are obtained by applying the rotation matrix

$$[\boldsymbol{C}_{\mathcal{R},\mathcal{I}}] = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.10}$$

where the subscript indicates the target reference frame, followed by the original frame from left to right. The positions of the primaries in the rotating frame are then computed as

$$\begin{aligned} \boldsymbol{r}_1 &= [\boldsymbol{C}_{\mathcal{R},\mathcal{I}}]\,\boldsymbol{R}_1 = -\mu\hat{\boldsymbol{x}} + 0\hat{\boldsymbol{y}} + 0\hat{\boldsymbol{z}} \\ \boldsymbol{r}_2 &= [\boldsymbol{C}_{\mathcal{R},\mathcal{I}}]\,\boldsymbol{R}_2 = (1-\mu)\hat{\boldsymbol{x}} + 0\hat{\boldsymbol{y}} + 0\hat{\boldsymbol{z}} \end{aligned} \tag{2.11}$$

where both B_1 and B_2 positions are fixed in $\mathcal{R}$. Finally, the position of the spacecraft with respect to the primaries is rewritten as

$$\begin{aligned} \boldsymbol{r}_{1,3} &= (x+\mu)\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}} \\ \boldsymbol{r}_{2,3} &= (x-1+\mu)\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}} \end{aligned} \tag{2.12}$$

With the definitions above, Equation 2.9 can be rewritten in the rotating frame considering that the angular acceleration of $\mathcal{R}$ with respect to $\mathcal{I}$ expressed in $\mathcal{R}$ is $\boldsymbol{\omega}_{\mathcal{R},\mathcal{I}} = 1\hat{\boldsymbol{z}}$ while the derivative of the position vector of the spacecraft in $\mathcal{R}$ is defined as $\boldsymbol{v}_3 = \dot{\boldsymbol{r}}_3 = \dot{x}\hat{\boldsymbol{x}} + \dot{y}\hat{\boldsymbol{y}} + \dot{z}\hat{\boldsymbol{z}}$. Therefore, the nondimensional equations of motion for the spacecraft in the Neptune-Triton rotating frame are written as

$$\begin{cases} \ddot{x} &= 2\dot{y} + x - \frac{(1-\mu)(x+\mu)}{r_{1,3}^3} - \frac{\mu(x-1+\mu)}{r_{2,3}^3} \\ \ddot{y} &= -2\dot{x} + y - \frac{(1-\mu)y}{r_{1,3}^3} - \frac{\mu y}{r_{2,3}^3} \\ \ddot{z} &= -\frac{(1-\mu)z}{r_{1,3}^3} - \frac{\mu z}{r_{2,3}^3} \end{cases} \tag{2.13}$$

From these equations, the pseudo-potential function U^* is defined as

$$U^* = \frac{1}{2}(x^2 + y^2) + \frac{(1-\mu)}{r_{1,3}} + \frac{\mu}{r_{2,3}} \tag{2.14}$$

Therefore, the equations of motion are simplified to

$$\begin{cases} \ddot{x} &= 2\dot{y} + \frac{\partial U^*}{\partial x} \\ \ddot{y} &= -2\dot{x} + \frac{\partial U^*}{\partial y} \\ \ddot{z} &= \frac{\partial U^*}{\partial z} \end{cases} \tag{2.15}$$

These equations do not have a generalized analytical solution; however, trajectories are computed from Equations 2.15 using numerical methods.

2.1.2 Jacobi Constant

An integral of motion exists in this autonomous dynamical system, labeled the Jacobi constant. This energy-like quantity exists because the CR3BP is an autonomous dynamical model [56]. The Jacobi constant is derived by calculating the derivative of the kinetic energy at B_3 in the rotating frame. Defining $\boldsymbol{v}_3$ as the value of the velocity of B_3 in the rotating frame, the derivative of the kinetic energy is

$$\frac{d}{dt}\Big(\frac{v_3^2}{2}\Big) = \frac{d}{dt}\Big(\frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)\Big) = \dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z} \tag{2.16}$$

Then, by substituting the expressions of the accelerations from Equation 2.15 in 2.16 and simplifying, the equation becomes

$$\frac{d}{dt}\Big(\frac{v_3^2}{2}\Big) = \dot{x}\frac{\partial U^*}{\partial x} + \dot{y}\frac{\partial U^*}{\partial y} + \dot{z}\frac{\partial U^*}{\partial z} = \frac{d}{dt}(U^*) \tag{2.17}$$

Since both sides of Equation 2.17 are derivatives in time, by integrating the equation and rearranging the terms, the Jacobi constant is obtained as [103]

$$C_J = 2U^* - \boldsymbol{v}_3 = 2U^* - \dot{x}^2 - \dot{y}^2 - \dot{z}^2 \tag{2.18}$$

The expression of the Jacobi constant suggests that it is inversely proportional to the total energy at B_3. For this reason, low values of the Jacobi constant correspond to high values of total energy and vice versa.

2.2 Fundamental Solutions

Although the CR3BP does not possess analytical solutions, fundamental solutions exist [103]. These fundamental solutions include equilibrium points, families of periodic and quasi-periodic orbits, and their hyperbolic invariant manifolds. Aside from providing a deeper understanding of the underlying dynamics, fundamental solutions can be leveraged in trajectory design to construct complex trajectories that exploit the natural transport mechanisms of the system [56]. These solutions are often computed numerically from local linear approximations of the dynamics. Due to the linearization, some solutions, such as periodic orbits, require correction in the nonlinear dynamics using correction schemes, such as shooting methods [103]. This section provides an overview of the definitions and computational approaches of the fundamental solutions of the Neptune-Triton CR3BP, which are leveraged in the motion-primitive informed trajectory design approach.

2.2.1 Equilibrium Points

The CR3BP admits five stationary points, where the time derivatives of the position are equal to zero in the rotating frame. Consequently, if not perturbed, an object placed in such points will maintain that position indefinitely [103, 56]. These points are commonly referred to as equilibrium or Lagrange points and are defined in the rotating frame. Specifically, there are five equilibrium points, usually labeled L_1, L_2, L_3, L_4, and L_5. The first three, L_1, L_2, and L_3, are identified as the collinear points as they lie along the $\hat{\boldsymbol{x}}$ direction of the rotating frame. While L_4 and L_5 are known as triangular points because they are located at the vertices of equilateral triangles that have the two primary bodies as the other two vertices [103, 56].

The values of these equilibrium points are computed starting from the Equations 2.15, by considering the definition of a stationary point $\boldsymbol{r}_{eq} = [x_{eq}, y_{eq}, z_{eq}]$, i.e. $\frac{d}{dt}(\boldsymbol{r}_{eq}) = \boldsymbol{0}$ and $\frac{d^2}{dt^2}(\boldsymbol{r}_{eq}) = \boldsymbol{0}$. By imposing these conditions on the equations of motion in the CR3BP, the following equations are obtained

$$\begin{aligned}
\ddot{x}_{eq} &= \frac{\partial U^*}{\partial x} = x_{eq} - \frac{(1-\mu)(x_{eq}+\mu)}{r_1^3} - \frac{\mu(x_{eq}-1+\mu)}{r_2^3} = 0 \\
\ddot{y}_{eq} &= \frac{\partial U^*}{\partial y} = y_{eq} - \frac{(1-\mu)y_{eq}}{r_1^3} - \frac{\mu y_{eq}}{r_2^3} = 0 \\
\ddot{z}_{eq} &= \frac{\partial U^*}{\partial z} = \frac{(1-\mu)z_{eq}}{r_1^3} - \frac{\mu z_{eq}}{r_2^3} = 0
\end{aligned} \tag{2.19}$$

This system of equations has a trivial solution, $z_{eq} = 0$, for every value of x and y, indicating that all the equilibrium points lie in the $x - y$ plane of the rotating frame. Then, the remaining two equations are rewritten as

$$\begin{aligned}
&x_{eq} - \frac{(1-\mu)(x_{eq}+\mu)}{r_1^3} - \frac{\mu(x_{eq}-1+\mu)}{r_2^3} = 0 \\
&y_{eq}\Big(1 - \frac{(1-\mu)}{r_1^3} - \frac{\mu}{r_2^3}\Big) = 0
\end{aligned} \tag{2.20}$$

As a consequence of the second Equation in 2.20, y can assume the value of $y_{eq} = 0$ or $y_{eq} \neq 0$.

Considering the case at $y_{eq} = 0$, the first equation in 2.20 is used to compute the values of the x-coordinate of the collinear points L_1, L_2 and L_3 by finding the roots of the function

$$f(x) = x_{eq} - \frac{(1-\mu)(x_{eq}+\mu)}{(\sqrt{(x_{eq}+\mu)^2})^3} - \frac{\mu(x_{eq}-1+\mu)}{(\sqrt{(x_{eq}+\mu-1)^2})^3} \tag{2.21}$$

Since Equation 2.21 is nonlinear, numerical approaches such as Newton's method can be used to find its roots. These approaches require an initial condition $\boldsymbol{x}_{s0}$, which is updated iteratively until the difference between the values at subsequent iterations is less than a predefined tolerance. By imposing Equation 2.21 equal to zero, three regions of the solution space are identified to provide a reasonable initial guess for the root-finding algorithm

$$\begin{aligned}
&(x_{eq} + \mu) > 0 \land (x_{eq} + \mu - 1) > 0 \\
&(x_{eq} + \mu) > 0 \land (x_{eq} + \mu - 1) < 0 \\
&(x_{eq} + \mu) < 0 \land (x_{eq} + \mu - 1) < 0
\end{aligned} \tag{2.22}$$

Consequently, the value of $\boldsymbol{x}_{s0}$ is computed from each interval as

$$\begin{aligned}
&x_{eq} > 1 - \mu \\
&-\mu < x_{eq} < 1 - \mu \\
&x_{eq} < -\mu
\end{aligned} \tag{2.23}$$

Once one value for each interval is selected, the three solutions for the x-coordinate of the collinear points are computed as the roots of Equation 2.21.

The positions of the triangular points L_4 and L_5 are computed considering the conditions $y_{eq} \neq 0$. In this case, the Equations 2.21 is rewritten as

$$\begin{aligned}
f(x) &= x_{eq}\Big(1 - \frac{(1-\mu)}{r_1^3} - \frac{\mu}{r_2^3}\Big) - \frac{\mu(1-\mu)}{r_1^3} + \frac{\mu(1-\mu)}{r_2^3} = 0 \\
f(y) &= y_{eq}\Big(1 - \frac{(1-\mu)}{r_1^3} - \frac{\mu}{r_2^3}\Big) = 0
\end{aligned} \tag{2.24}$$

which are satisfied when $r_1 = r_2 = 1$, i.e., when the distance between a state and the two primaries is equal. From this condition, it is possible to derive the solutions to Equations 2.24 from the position of B_1 and B_2 in the rotating frame. Therefore

$$\begin{gathered}
\sqrt{(x_{eq} - (-\mu))^2 + y_{eq}^2} = 1 \ \land \ \sqrt{(x_{eq} - (1-\mu))^2 + y_{eq}^2} = 1 \\
\rightarrow x_{eq} = 0.5 - \mu \ \land \ y_{eq} = \pm - \sqrt{3}/2
\end{gathered} \tag{2.25}$$

Commonly, L_4 is assigned to the point with positive $\hat{\boldsymbol{y}}$ component and L_5 to the point with negative $\hat{\boldsymbol{y}}$ component.

The truncated position vector at the equilibrium points is listed in Table 2.1. The positions of the equilibrium points in the rotating frame in the NT-CR3BP are also represented in Figure 2.3. In this plot, the equilibrium points are identified with red diamonds, while the primaries are labeled N and T and are represented as blue and black dots, respectively.

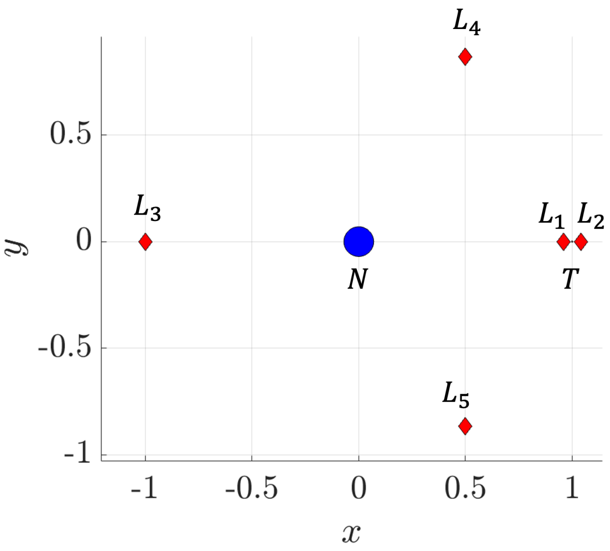


Figure 2.3: Location of the equilibrium points in the rotating frame in the NT-CR3BP

Table 2.1: Location of the equilibrium points in the Neptune-Triton CR3BP

Libration Point	x	y
L_1	0.959217	0
L_2	1.041493	0
L_3	-1.000087	0
L_4	0.499791	0.866025
L_5	0.499791	-0.866025

2.2.2 Periodic Orbits

Periodic orbits are fundamental solutions of the CR3BP that repeat their path after a specific time, known as the period. This condition is true for every state $\boldsymbol{x}(t)$ along the trajectory and can be described as

$$\boldsymbol{x}(t_1) = \boldsymbol{x}(t_1 + T) \tag{2.26}$$

where T is the period of the orbit. Periodic orbits are typically employed as target conditions for trajectory design [56] and are also utilized in this work to extract motion primitives that describe the dynamics governing the system.

2.2.2.1 Libration Point Orbits

A variety of periodic orbit families are generally found from a linear stability analysis of the nearby equilibrium point [103] using the variational equations. These orbits are typically classified according to the libration point from which they are generated. The variational equations describing the linearized perturbed motion around the equilibrium points L_1, L_2, L_3, L_4, and L_5 are presented. The dynamics linearized about the equilibrium points are described by the following equations:

$$\begin{aligned} \ddot{\xi} - 2\dot{\eta} &= U^*_{xx}|_{\vec{x}_{eq}}\xi + U^*_{xy}|_{\vec{x}_{eq}}\eta \\ \ddot{\eta} + 2\dot{\xi} &= U^*_{yx}|_{\vec{x}_{eq}}\xi + U^*_{yy}|_{\vec{x}_{eq}}\eta \\ \ddot{\zeta} &= U^*_{zz}|_{\vec{x}_{eq}}\zeta \end{aligned} \tag{2.27}$$

where ξ, η, ζ are the variations relative to the equation points in the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ and $\hat{\hat{z}}$ axes of the rotating frame in Equation 2.15, and $U^*_{xx}|_{\vec{x}_{eq}}, U^*_{yy}|_{\vec{x}_{eq}}, U^*_{xy}|_{\vec{x}_{eq}}$ are the second order partial derivatives of the pseudo-potential function evaluated at the equilibrium points [103]. From the matrix form of the variational equations describing the linear motion along the $\hat{\boldsymbol{x}}$- and $\hat{\boldsymbol{y}}$-axes, it is possible to compute the characteristic equation:

$$0 = \lambda^4 + \lambda^2(4 - U^*_{xx}|_{\vec{x}_{eq}} - U^*_{yy}|_{\vec{x}_{eq}}) + (U^*_{xx}|_{\vec{x}_{eq}} U^*_{yy}|_{\vec{x}_{eq}} - U^*_{xy}|^2_{\vec{x}_{eq}}) \tag{2.28}$$

The four in-plane eigenvalues are calculated by setting and applying the quadratic equation:

$$\Lambda = \frac{-4 + U^*_{xx}|_{\vec{x}_{eq}} + U^*_{yy}|_{\vec{x}_{eq}}}{2} \pm \frac{\sqrt{(4 - U^*_{xx}|_{\vec{x}_{eq}} - U^*_{yy}|_{\vec{x}_{eq}})^2 - 4(U^*_{xx}|_{\vec{x}_{eq}} U^*_{yy}|_{\vec{x}_{eq}} - U^*_{xy}|^2_{\vec{x}_{eq}})}}{2} \tag{2.29}$$

The four solutions are $\lambda_1 = +\sqrt{(\Lambda_1)}$, $\lambda_2 = -\sqrt{(\Lambda_1)}$, $\lambda_3 = +\sqrt{(\Lambda_2)}$ and $\lambda_4 = -\sqrt{(\Lambda_2)}$. The out-of-plane modes are found by solving the second-order differential equation linked to the variation $\hat{\boldsymbol{z}}$. The solution to this equation gives the two following eigenvalues $\lambda_5 = +i\sqrt{|U^*_{zz}|_{\vec{x}_{eq}}|}$ and $\lambda_6 = -i\sqrt{|U^*_{zz}|_{\vec{x}_{eq}}|}$, which are both linked to oscillatory modes for all equation points in the CR3BP [103].

Given the μ value for the system under investigation, the values of the in-plane and out-of-plane eigenvalues are computed at each equilibrium point. From the Lyapunov center theorem, the presence of oscillatory modes at an equilibrium point indicates the presence of periodic orbits emanating from that point [56]. Initial conditions for a periodic orbit are found by solving the Equations 2.27. In particular, a planar periodic orbit is computed by exciting the oscillatory modes of the first two equations in 2.27.

The solutions of the planar component of the variational equations in 2.27 are

$$\begin{aligned} \xi_0 &= \sum_{j=1}^{4} A_j e^{\lambda_j t_0} \ , \ \dot{\xi_0} = \sum_{j=1}^{4} \lambda_j A_j e^{\lambda_j t_0} \\ \eta_0 &= \sum_{j=1}^{4} B_j e^{\lambda_j t_0} \ , \ \dot{\eta_0} = \sum_{j=1}^{4} \lambda_j B_j e^{\lambda_j t_0} \end{aligned} \tag{2.30}$$

The coefficients A_j and B_j are coupled, and their relationship is found by substituting the solutions into the first variational equation 2.27 as

$$\sum_{j=1}^{4} (\lambda_j^2 A_j e^{\lambda_j t_0} - 2\lambda_j B_j e^{\lambda_j t_0}) = \sum_{j=1}^{4} (U^*_{xx}|_{\vec{x}_{eq}} A_j e^{\lambda_j t_0} + U^*_{xy}|_{\vec{x}_{eq}} B_j e^{\lambda_j t_0}) \tag{2.31}$$

Dividing by $e^{\lambda_j t_0}$ and rearranging

$$\sum_{j=1}^{4} A_j (\lambda_j^2 - U^*_{xx}|_{\vec{x}_{eq}}) = \sum_{j=1}^{4} B_j (U^*_{xy}|_{\vec{x}_{eq}} - 2\lambda_j) \tag{2.32}$$

Then, following the same procedure for the second equation in 2.27

$$\sum_{j=1}^{4} B_j (\lambda_j^2 - U^*_{yy}|_{\vec{x}_{eq}}) = \sum_{j=1}^{4} A_j (U^*_{yx}|_{\vec{x}_{eq}} - 2\lambda_j) \tag{2.33}$$

The equations in 2.32 and 2.33 are equivalent, and from the first one, it is possible to derive the relation

$$B_j = \frac{A_j (\lambda_j^2 - U^*_{xx}|_{\vec{x}_{eq}})}{2\lambda_j + U^*_{xy}|_{\vec{x}_{eq}}} = \alpha_j A_j \tag{2.34}$$

Where $U^*_{xy}|_{\vec{x}_{eq}} = 0$ at the collinear points. Then substituting 2.34 in 2.30, the solutions become:

$$\begin{aligned}
\xi_0 &= \sum_{j=1}^{4} A_j e^{\lambda_j t_0} = A_1 e^{\lambda_1 t_0} + A_2 e^{\lambda_2 t_0} + A_3 e^{\lambda_3 t_0} + A_4 e^{\lambda_4 t_0} \\
\dot{\xi_0} &= \sum_{j=1}^{4} \lambda_j A_j e^{\lambda_j t_0} = \lambda_1 A_1 e^{\lambda_1 t_0} + \lambda_2 A_2 e^{\lambda_2 t_0} + \lambda_3 A_3 e^{\lambda_3 t_0} + \lambda_4 A_4 e^{\lambda_4 t_0} \\
\eta_0 &= \sum_{j=1}^{4} \alpha_j A_j e^{\lambda_j t_0} = \alpha_1 A_1 e^{\lambda_1 t_0} + \alpha_2 A_2 e^{\lambda_2 t_0} + \alpha_3 A_3 e^{\lambda_3 t_0} + \alpha_4 A_4 e^{\lambda_4 t_0} \\
\dot{\eta_0} &= \sum_{j=1}^{4} \lambda_j \alpha_j A_j e^{\lambda_j t_0} = \lambda_1 \alpha_1 A_j e^{\lambda_1 t_0} + \lambda_2 \alpha_2 A_j e^{\lambda_2 t_0} + \lambda_3 \alpha_3 A_j e^{\lambda_3 t_0} + \lambda_4 \alpha_4 A_j e^{\lambda_4 t_0}
\end{aligned} \tag{2.35}$$

By solving the system of equations, it is possible to find A_1, A_2, A_3, and A_4 given a set of initial conditions, or to choose initial conditions that excite specific modes. The oscillatory modes are excited when $A_1 = A_2 = 0$, if λ_1 and λ_2 correspond to the real eigenvalues. In this scenario, the initial values for $\dot{\xi_0}$ and $\dot{\eta_0}$ are obtained as

$$\dot{\xi_0} = \frac{\eta_0 \lambda_3}{\alpha_3}, \quad \dot{\eta_0} = \xi_0 \alpha_3 \lambda_3 \tag{2.36}$$

Considering the values of the pseudo-potential and the eigenvalues at the collinear points, it is possible to find the initial conditions for a planar periodic orbit by considering a small perturbation ξ or η about the position of the equilibrium point [103].

2.2.2.2 Resonant Orbits

Resonant orbits are defined based on the two-body problem (2BP) [78, 107]. In this context, a spacecraft is said to follow a p_r:q_r resonant orbit if it completes p_r revolutions around the main primary body in the same amount of time that the smaller primary completes q_r revolutions around the main primary in the inertial frame [78]. The resonance is termed interior if $p_r > q_r$, and exterior when $q_r > p_r$. Depending on the direction of angular momentum, the orbit is prograde or retrograde, aligning with the positive or negative direction of the inertial frame's $\hat{\boldsymbol{z}}$-axis, respectively. The initial condition for a resonant orbit is obtained by defining the orbital elements of an elliptical orbit in the 2BP framework. First, the semi-major axis is computed as

$$a = \left(\frac{q_r}{p_r}\right)^{2/3} a_m \tag{2.37}$$

where a_m is the semi-major axis of B_2's orbit. The eccentricity e is then selected within the range $[0.2, 0.8]$ to ensure that the apses remain sufficiently distant from the primaries. Once a and e are defined, the position and velocity at one of the apses are calculated as

$$\begin{aligned} r_a &= a(1 \pm e) \\ v_a &= \sqrt{\mu_N \left(\frac{2}{r_a} - \frac{1}{a}\right)} \end{aligned} \tag{2.38}$$

Here, $\mu_N = \tilde{G}\tilde{M}_N = 6.8351 \cdot 10^6$ km^3/s^2 is Neptune's gravitational parameter [82]. The minus sign yields the periapsis position, while the plus sign gives the apoapsis. The inertial frame is centered on B_1, with its axes aligned with those of the CR3BP rotating frame at a chosen time t, such that both frames are instantaneously aligned as represented in Figure 2.4. In this frame, the apsis is placed along the $\hat{\boldsymbol{X}}$-axis and defined as

$$\boldsymbol{r}_a = [\pm r_a, 0, 0]^T \tag{2.39}$$

Selecting the sign of the x-component allows to capture different geometries corresponding to the same p_r:q_r resonance. For families symmetric about both the x and y-axes, additional configurations can be generated by placing the apsis along the $\hat{\boldsymbol{Y}}$-axis instead:

$$\boldsymbol{r}_a = [0, r_a, 0]^T \tag{2.40}$$

The velocity at the apsis is perpendicular to the position vector, yielding:

$$\boldsymbol{v}_a = [0, \pm v_a, 0]^T \tag{2.41}$$

A positive sign produces a prograde orbit in the inertial frame, while a negative sign results in an orbit with retrograde direction of motion, as it is represented in Figure 2.4 for a 1:3 resonant orbit with a) prograde and b) retrograde initial conditions. When the apsis lies along the $\hat{\boldsymbol{Y}}$-axis, the velocity vector becomes:

$$\boldsymbol{v}_a = [\pm v_a, 0, 0]^T \tag{2.42}$$

To obtain the resonant orbit in the CR3BP, the initial condition is first transformed into the rotating frame, then propagated using the CR3BP equations of motion, and finally corrected to enforce periodicity. Depending on the placement of the initial condition along the $\hat{\boldsymbol{X}}$-axis of the inertial reference frame and its direction of motion, the resonant orbit is defined with the label $\pm x, \pm h$, where $+x$ $(-x)$ indicates the placement along the positive (negative) side of the $\hat{\boldsymbol{X}}$-axis and $+h$ $(-h)$ indicates a prograde (retrograde) motion.

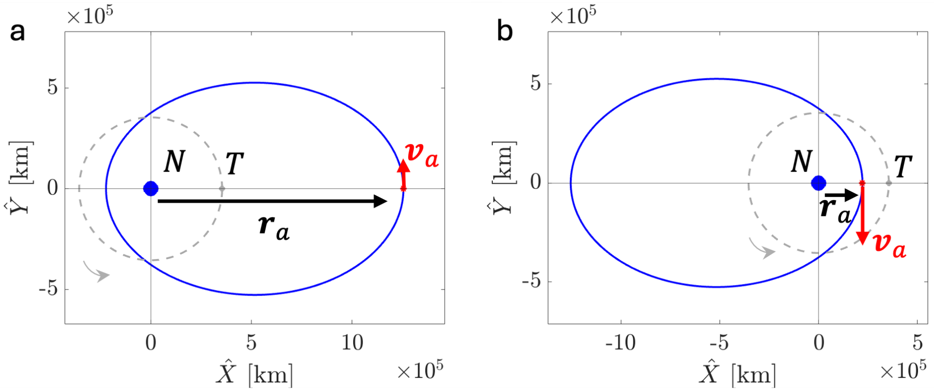


Figure 2.4: Examples of initial condition computation for a planar 1:3 resonant orbit in the Neptune 2BP: a) using the apoapsis on $+\hat{\boldsymbol{X}}$ to generate a prograde orbit $+h$ and b) using the periapsis on $+\hat{\boldsymbol{X}}$ to generate a retrograde orbit $-h$

2.2.2.3 B_2-Centered Orbits

Periodic orbits that evolve around the smaller primary are referred to as the B_2-centered orbits. This type of orbit includes the distant prograde (DPO) and retrograde orbit families (DRO), as well as the low prograde orbits (LPO), which are often leveraged in trajectory designs targeting the exploration of the system's second primary. In this work, the DPO family and its hyperbolic invariant manifolds are leveraged in the design of a trajectory to reach a low prograde orbit around Triton.

B_2-centered orbits can be generated from analytical approximations of the dynamics around the B_2 as presented by Hénon [49], or with numerical approaches [51]. In the analytical approach, for the planar B_2-centered orbits, the derivation considers the first two equations of motion for the

CR3BP, as given in Equation 2.13. In the case that $r_{2,3} \ll 1$ and $\mu \to 0$, i.e., the spacecraft is very close to the second primary and the mass of the first primary is much larger than the mass of the second primary, the Equations 2.13 are simplified by expanding the second term on the left-hand side around $x = y = 0$ and neglecting the seond order terms of small quantities to obtain

$$\begin{aligned} \ddot{x} &= 2\dot{y} + 3x - \frac{x}{\rho^3} \\ \ddot{y} &= 2\dot{x} - \frac{y}{\rho^3} \end{aligned} \tag{2.43}$$

with

$$\rho = \sqrt{x^2 + y^2} \tag{2.44}$$

which are referred to as the Hill's equations. With these equations, a new constant of motion is found from the expression of the Jacobi constant and is defined as

$$\Gamma = 3x^2 + \frac{2}{\rho} - \dot{x}^2 - \dot{y}^2 \tag{2.45}$$

Considering this dynamics, a periodic orbit can be defined by the value of the constant Γ and of the variable x. In particular, for an orbit that crosses the $\hat{\boldsymbol{x}}$-axis perpendicularly in the positive direction, the state has $y = 0$ and $\dot{x} = 0$, thus the value of $\dot{y}$ can be found from the knowledge of x and Γ. Therefore, an orbit can be represented as a point in an $x - \Gamma$ plane, and a periodic orbit family is a curve in that space [49]. At this point, initial conditions can be found from a grid search in the x and Γ space, or considering previously published values [50, 55, 53, 69]. The initial conditions can be integrated with the CR3BP equations and corrected for periodicity; then, a family can be found using a continuation scheme, such as the pseudo-arclength continuation method.

2.2.2.4 Stability of a Periodic Orbit

The variational equations can also be used to study the stability at any state along a periodic orbit in the CR3BP. The stability evaluation is an important step in trajectory design as it supplies insight into the dynamics governing the regions around the selected orbit. For example, if the orbit is unstable, a small perturbation of the spacecraft's state might lead to departure from the orbit, requiring frequent station-keeping maneuvers. Therefore, considering a state along a periodic orbit $\boldsymbol{x}(t)$ with an orbit period T, the system of variational equations evaluated at $\boldsymbol{x}(t)$ is a linear time-varying system [56]. The first order linear approximation of the evolution of the perturbed state along a period orbit, $\boldsymbol{x}(t)$, after a period T is

$$\delta\boldsymbol{x}(t+T) = \boldsymbol{\Phi}(t+T,t)\delta\boldsymbol{x}(t) \tag{2.46}$$

where $\boldsymbol{\Phi}(t+T,t)$ is the monodromy matrix of the periodic orbit, defined as the state transition matrix (STM) propagated for one orbital period [56]

$$\boldsymbol{M}(t) = \boldsymbol{\Phi}(t+T,t) \tag{2.47}$$

By leveraging the properties of the STM, the value of the monodromy matrix is defined at any other t_{i+1} along the periodic orbit as

$$\boldsymbol{M}(t_{i+1}) = \boldsymbol{\Phi}(t_i,t_{i+1})^{-1}\boldsymbol{M}(t_i)\boldsymbol{\Phi}(t_i,t_{i+1}) \tag{2.48}$$

proving that the monodromy matrices at two different states along a periodic orbit are similar, and therefore have the same stability properties [14].

The stability of a periodic orbit is assessed from the eigenvalues of its monodromy matrix at any state. $\boldsymbol{M}$ is a real matrix; additionally, due to the autonomous Hamiltonian structure of the dynamical system, $\boldsymbol{M}$ is also a symplectic matrix [56]. As a consequence of these properties, the eigenvalues of $\boldsymbol{M}$ exist in reciprocal pairs, and all complex eigenvalues exist in complex conjugate pairs [56]. Moreover, $\boldsymbol{M}$ admits a pair of trivial eigenvalues always equal to 1 [56], which are associated with the dynamical structure of the periodic orbit and its existence in a family. On the other hand, the other eigenvalues are used to assess the stability of the orbit.

The eigenvalues of the monodromy matrix reflect the mode resulting from stepping along the relative eigenvector. The modes are deducted from the eigenvalues as in any discrete linear time-invariant system, by their location relative to the unit circle in the complex plane. Therefore, when the eigenvalue is $\lambda = a$ and $|\lambda| < 1$, the associated mode is stable; otherwise, if $|\lambda| > 1$, the mode is unstable. On the other hand, if the eigenvalue is complex, i.e., $\lambda = a + bj$, then the associated mode is oscillatory if $|\lambda| = 1$, stable spiral if $|\lambda| < 1$, and unstable spiral if $|\lambda| > 1$.

Since the eigenvalues exist in reciprocal pairs, computing the sum of the non-trivial reciprocal eigenvalue pairs defines the stability indices s_1 and s_2 [52]. For planar periodic orbits, the value of s_1 indicates the in-plane orbit modes while s_2 indicates the out-of-plane modes. However, this distinction is not possible for spatial orbits, where s_1 and s_2 are arbitrarily selected for one family member and maintained consistently along the family. If s_1 and s_2 lie within the range $[-2, 2]$, then the orbit is linearly stable and the motion in its surroundings is bounded. On the other hand, if either of the two stability indices is outside the $[-2, 2]$ boundaries, the orbit is considered unstable and possesses hyperbolic invariant manifolds.

2.2.2.5 Correction via Multiple Shooting

A multiple-shooting scheme is an iterative numerical method that solves multiple two-boundary value problems, which are coupled by continuity constraints [56]. This approach is often used while correcting periodic orbits or trajectories in the CR3BP, as discretizing such a complex two-boundary value problem into several smaller ones facilitates the correction process. This numerical approach works by creating a vector of states and times, describing the states along a trajectory, and then updating it with Newton's method until a set of constraints is satisfied [56, 8].

In this dissertation, a free variable vector is constructed using states equally spaced in time along the discontinuous initial guess of a periodic orbit, and the time constant interval between two samples. Starting from the initial condition of the initial guess, the samples capture the initial state of each arc along the trajectory that constitutes the first states of a two-point boundary value problem. Thus, considering the desired number of arcs along the initial guess n_a, the free variable vector is computed as

$$\boldsymbol{V} = [\boldsymbol{x}_{1,i}^T, \boldsymbol{x}_{2,i}^T, ..., \boldsymbol{x}_{n_a,i}^T, \Delta t]^T \tag{2.49}$$

where $\boldsymbol{x}_{j,i}$ identifies the state at the beginning of the jth arc, and $\Delta t = T/n_a$. This vector has dimension $n_V = 6n_a + 1$.

The constraint vector is formulated to ensure continuity between successive arcs and between the initial state of the first arc and the final state of the final arc. Thus, the constraint between consecutive arcs is

$$\boldsymbol{F}_c = [\boldsymbol{x}_{1,f} - \boldsymbol{x}_{2,i}]^T \tag{2.50}$$

While the periodicity constraint is formulated as

$$\boldsymbol{F}_p = [\boldsymbol{x}^{(5)}_{n_a,f} - \boldsymbol{x}^{(5)}_{1,i}]^T \tag{2.51}$$

where the superscript (5) above each state indicates that the constraint is only imposed on the x, y, z, $\dot{x}$ and $\dot{z}$ component at those states, since the value of $\dot{y}$ is implicitly enforced by conservation of the Jacobi constant value along a natural solution. Finally, a constraint is imposed on the value of y at the initial state to avoid ambiguity regarding the location of the initial state and facilitate the correction process. Therefore, the last component of the constraint vector is

$$F_{y0} = y_{1,1} - y_d \tag{2.52}$$

Where y_d is a user-defined value, based on the type of periodic orbit. Overall, the constraint vector is defined as

$$\boldsymbol{F}(\boldsymbol{V}) = [\boldsymbol{F}_{c,1}, \boldsymbol{F}_{c,2}, ..., \boldsymbol{F}_{c,n_a}, \boldsymbol{F}_p, F_{y0}]^T \tag{2.53}$$

where the $\boldsymbol{F}(\boldsymbol{V})$ has dimension $n_F = 6(n_a - 1)$.

The free variable vector is corrected iteratively until the constraints are met. First, each node of the feature vector is propagated with the dynamics described in Section 2.1. Then, after propagating each state for the corresponding Δt, each component of the constraint vector is evaluated, and the norm of the total vector is computed. If $||\boldsymbol{F}(\boldsymbol{V})|| \neq 0$, then the free variable vector must be updated. Due to possible inaccuracies of numerical propagation, it might not be possible to achieve the values such that $||\boldsymbol{F}(\boldsymbol{V})|| = 0$, therefore, the termination condition is defined as $||\boldsymbol{F}(\boldsymbol{V})|| < tol$. In this work, the tolerance is set to $tol = 10^{-12}$. If this condition is not met, then the free variable vector is updated using Newton's method [56, 8]. Considering the free variable vector $\boldsymbol{V}^*$ that satisfies the condition $\boldsymbol{F}(\boldsymbol{V}^*) = 0$, a vector $\boldsymbol{V}^*$ close to the initial guess should fulfill the condition

$$\boldsymbol{F}(\boldsymbol{V}^*) = 0 \approx \boldsymbol{F}(\boldsymbol{V}) + D\boldsymbol{F}(\boldsymbol{V})(\boldsymbol{V}^* - \boldsymbol{V}) \tag{2.54}$$

where D$\boldsymbol{F}(\boldsymbol{V})$ is the Jacobian of $\boldsymbol{F}$ with respect to the free variable vector $\boldsymbol{V}$ and has dimensions $n_F \times n_V$. From Equation 2.54, two update equations are written as

$$\begin{aligned} \boldsymbol{V}_{i+1} &= \boldsymbol{V}_i - D\boldsymbol{F}(\boldsymbol{V}_i)^T[D\boldsymbol{F}(\boldsymbol{V}_i) \cdot D\boldsymbol{F}(\boldsymbol{V}_i)^T]^{-1}\boldsymbol{F}(\boldsymbol{V}_i) \quad \text{if } n_V > n_F \\ \boldsymbol{V}_{i+1} &= \boldsymbol{V}_i - D\boldsymbol{F}(\boldsymbol{V}_i)^{-1}\boldsymbol{F}(\boldsymbol{V}_i) \quad \text{if } n_V = n_F \end{aligned} \tag{2.55}$$

In both cases, the $D\boldsymbol{F}(\boldsymbol{V}_i)$ matrix is computed with the analytical formulation of the partial derivatives as

$$D\boldsymbol{F}(\boldsymbol{V}_i) = \left[\frac{\partial \boldsymbol{F}_{c,1}}{\partial \boldsymbol{V}}, \frac{\partial \boldsymbol{F}_{c,2}}{\partial \boldsymbol{V}}, ..., \frac{\partial \boldsymbol{F}_{c,n_a}}{\partial \boldsymbol{V}}, \frac{\partial \boldsymbol{F}_{y0}}{\partial \boldsymbol{V}}\right]^T \tag{2.56}$$

where each component is computed as

$$\frac{\partial \boldsymbol{F}_{c,i}}{\partial \boldsymbol{V}} = [0_{6\times6(i-1)},\ -\boldsymbol{\Phi}(t_{i+1}, t_i),\ \boldsymbol{I}_{6\times6},\ \mathbf{0}_{6\times(6(n_a-1)-6)},\ \dot{\boldsymbol{x}}_{i,f}],\ i \neq n_a \tag{2.57}$$

and

$$\frac{\partial \boldsymbol{F}_{c,n_a}}{\partial \boldsymbol{V}} = [\boldsymbol{I}^{(5)}_{6\times6},\ \mathbf{0}^{(5)}_{6\times6(n_a-12)},\ -\boldsymbol{\Phi}(t_{i+1}, t_i)^{(5)},\ \dot{\boldsymbol{x}}^{(5)}_{i,f}] \tag{2.58}$$

and

$$\frac{\partial \boldsymbol{F}_{y0}}{\partial \boldsymbol{V}} = [0,\ 1,\ \mathbf{0}_{6\times6(n_a-1)}] \tag{2.59}$$

Once Equation 2.57 is evaluated for all the arcs, then these expressions are used to compute the $D\boldsymbol{F}(\boldsymbol{V})$ matrix and the value of $\boldsymbol{V}$ is updated until $||\boldsymbol{F}(\boldsymbol{V})|| < tol$.

2.2.2.6 Pseudo-Arclength Continuation

Periodic orbits exist in families, and they are computed using numerical methods such as pseudo-arclength continuation [54]. This continuation approach involves using one or more variables of a family member to determine the initial conditions of the next one. These variables are perturbed by stepping along the function describing their variation within the family. The pseudo-arclength continuation leverages the orbit's representation with its free variable vector $\boldsymbol{V}$ across the family.

When a member of the family p_j is known and is represented with $\boldsymbol{V}_j$, the initial conditions of the next orbit are computed by finding the instantaneous rate of change of the solution curve, to find a first-order approximation of $\boldsymbol{V}_{j+1}$. This new free variable vector is then used within a multiple-shooting scheme to correct the initial guess for the new member of the periodic orbit family and find $\boldsymbol{V}_{j+1}$ [54, 8].

Given the definition of the free variable vector $\boldsymbol{V}$, the constraint vector $\boldsymbol{F}$ and the Jacobian $D\boldsymbol{F}(\boldsymbol{V})$ from Section 2.2.2.5, the matrix $D\boldsymbol{F}(\boldsymbol{V})$ has a 1-dimensional null space only if the condition $|n_F - n_V| = 1$ is met. Therefore, the function $\boldsymbol{F}(\boldsymbol{V}) = 0$ has infinite solutions, i.e., there are infinite free variable vectors that can satisfy the constraint. From this property, the instantaneous variation of the curve is computed as the null-space of $D\boldsymbol{F}(\boldsymbol{V})$. Thus, given the free variable vector of a periodic orbit of the family $\boldsymbol{V}_j$, the initial guess for the next periodic orbit is found as

$$\boldsymbol{V}_{j+1} = \boldsymbol{V}_j + \delta_s \; Null(D\boldsymbol{F}(\boldsymbol{V})) \tag{2.60}$$

where δ_s is a user-defined scalar value. Given the initial guess of $\boldsymbol{V}_{j+1}$, the correction process to recover the new periodic orbit of the family is similar to the one described in Section 2.2.2.5. A modification is employed in the definition of the constraint vector, to ensure that the correct value of $\boldsymbol{V}_{j+1}$ lies within a step δ_s from $\boldsymbol{V}_j$ along the direction given by $Null(D\boldsymbol{F}(\boldsymbol{V}))$. The new constraint vector $\boldsymbol{H}(\boldsymbol{V})$ is then defined as

$$\boldsymbol{H}(\boldsymbol{V}) = [\boldsymbol{F}(\boldsymbol{V}),\; (\boldsymbol{V}_{j+1} - \boldsymbol{V}_j)^T (Null(D\boldsymbol{F}(\boldsymbol{V})) - \delta_s)]^T \tag{2.61}$$

Other continuation schemes exist, such as natural parameter continuation, where the continuation is based on the perturbation of a natural property of an orbit, such as its period, Jacobi constant, or a component of its initial state. While the pseudo-arclength continuation might be computationally more expensive than a natural parameter continuation, it is preferred to the latter as it does not require prior knowledge on the evolution of a specific parameter within the family. Moreover, it is more robust to sudden variations in the free variable vector curve, enabling larger step sizes during the continuation process.

The pseudo-arclength continuation scheme is used to find any periodic orbit family used in this dissertation. Figure 2.5 shows an example of a libration point, a primary-centered, and a resonant periodic orbit family existing in the Neptune-Triton CR3BP. In particular, Figure 2.5a shows selected members of the L_1 Lyapunov orbit family, generated by the planar oscillatory mode obtained from linearized perturbed dynamics about L_1. The family is generated until the last orbit of the family reaches Triton's surface. Figure 2.5b shows some members of the distant prograde orbit family centered about Triton. The family is generated from initial conditions of a periodic moon-centered orbit provided by Dr. Natasha Bosanac, then propagated until an initial condition produces an orbit that intersects with Triton's surface. Finally, Figure 2.5c provides a representation of a 3:1 resonant orbits family. First, a single resonant orbit is generated from 3:1 resonant orbit defined in the Neptune 2BP with apoapsis on the $\hat{\boldsymbol{X}}$-axis and prograde direction of motion, and then corrected in the NT-CR3BP. Then, the family is obtained via pseudo-arclength continuation until intersection with Neptune's radius.

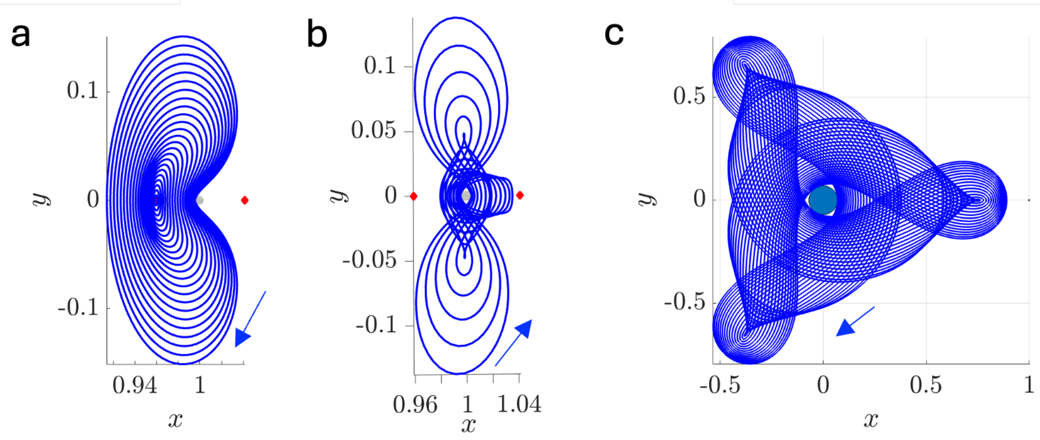


Figure 2.5: Families of libration periodic orbits in the Neptune-Triton System: a) L_1 Lyapunov orbits family, b) distant prograde orbits family, c) 3:1 resonant orbits family

2.2.3 Hyperbolic Invariant Manifolds

Hyperbolic invariant manifolds constitute another fundamental solution of the CR3BP. These structures appear when a periodic orbit is linearly unstable along one or more of the directions of the eigenvectors of its monodromy matrix. The hyperbolic invariant manifolds are classified as stable or unstable if they are respectively asymptotically approaching or departing from a periodic orbit as $t \to \infty$ [56, 103]. They are computed numerically by perturbing a state $\boldsymbol{x}(t)$ along the periodic orbit by a value γ, along the direction provided by the stable or unstable eigenvector of the monodromy matrix at that state as:

$$\begin{aligned} \boldsymbol{x}^s(t) &= \boldsymbol{x}(t) \pm \gamma \hat{\boldsymbol{v}}^s \\ \boldsymbol{x}^u(t) &= \boldsymbol{x}(t) \pm \gamma \hat{\boldsymbol{v}}^u \end{aligned} \tag{2.62}$$

where $\hat{\boldsymbol{v}}^s$ and $\hat{\boldsymbol{v}}^u$ are the unit vectors along the eigenvector $\boldsymbol{v}^s$ and $\boldsymbol{v}^u$. The perturbed state $\boldsymbol{x}(t)^s$ is propagated backward in time to obtain a trajectory along the stable half-manifold approaching the periodic orbit. Vice versa, $\boldsymbol{x}(t)^u$ is propagated forward in time to produce a trajectory along the unstable half-manifold departing the periodic orbit. The propagation time is typically selected based on the application of the manifold trajectories. Usually, trajectories along hyperbolic invariant manifolds are propagated until reaching a certain event, such as the crossing of a hypersurface or the impact with a primary.

Examples of hyperbolic invariant manifolds are presented in Figure 2.6. These manifolds are generated by stepping along the stable (blue) and unstable (red) eigenvectors obtained from the linearized perturbed dynamics at about 100 points along an L_1 Lyapunov orbit in the NT-CR3BP. Figure 2.6a represents the trajectories obtained from stepping along the positive direction of the eigenvectors in Equation 2.62, propagated until hitting the terminal conditions imposed as $x = 0.95$ and $x = 1.10$. On the other hand, Figure 2.6b shows the trajectories obtained by stepping along the negative direction of the eigenvectors and imposing the terminal condition at $x = 0$.

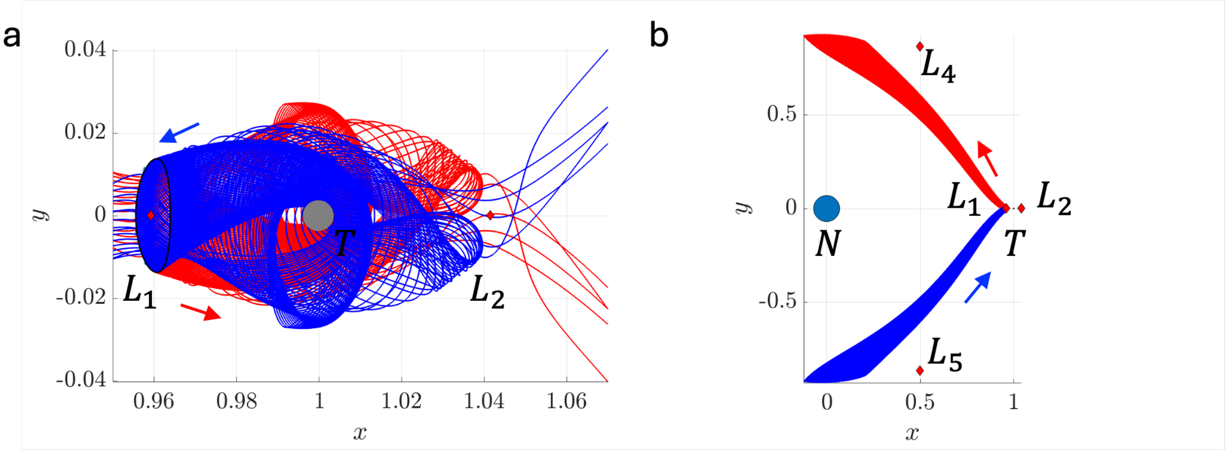


Figure 2.6: Example of stable (blue) and unstable (red) hyperbolic invariant manifolds generated from an L_1 Lyapunov orbit with $C_J = 3.01385$ and directed a) along the positive direction and b) along the negative direction of the stable and unstable eigenvectors

2.3 Point Mass Ephemeris Model

In Chapter 7, trajectories in the Neptunian system are corrected in an ephemeris model, and orbits in Chapter 5 are propagated in a high-fidelity gravity model around the Moon. In the first case, trajectories are obtained considering the point mass gravity of n bodies at each instant. On the other hand, lunar orbits are generated considering the high-fidelity gravity model of the Moon augmented by the point mass gravity of secondary bodies.

In this model, states are defined in an inertial frame centered on the primary body, e.g., the axes of the International Celestial Reference Frame centered on the primary body, as shown in Figure 2.7. The state vector for the spacecraft relative to the primary body is defined as $\tilde{\boldsymbol{X}} = [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}]^T = [\tilde{\boldsymbol{R}}_{B_i,3}, \tilde{\boldsymbol{V}}_{B_i,3}]^T$.

The equations of motion for a spacecraft subject to the gravitational influence of N_b bodies in an inertial reference frame are computed from Newtonian mechanics, considering the gravitational attraction force between two bodies, B_i and B_j defined as

$$\boldsymbol{F}_{i,j} = \frac{-\tilde{G}\tilde{M}_i\tilde{M}_j}{\tilde{R}_{i,j}^3}\tilde{\boldsymbol{R}}_{i,j} \tag{2.63}$$

where $\tilde{G}$ is the universal gravitational constant, $\tilde{M}_i$ and $\tilde{M}_j$ are the mass of B_i and B_j respectively,

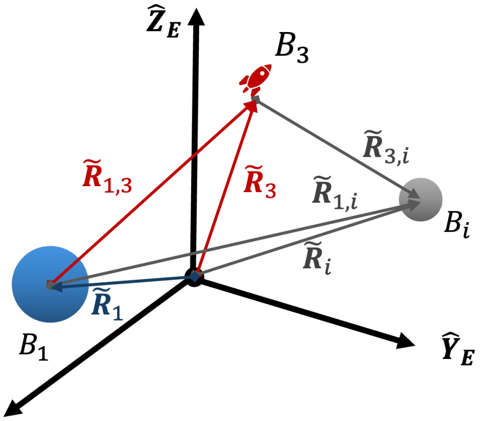


Figure 2.7: Configuration of B_1, B_3 and a ith celestial body in an inertial reference frame $\mathcal{I}_E$ with axis $\hat{\boldsymbol{X}}_E, \hat{\boldsymbol{Y}}_E, \hat{\boldsymbol{Z}}_E$

$\tilde{\boldsymbol{R}}_{i,j}$ is the position of B_j and B_i, and $\tilde{R}^3_{i,j}$ is the l^2-norm of that vector. Then, considering this force as acting on B_3, given the gravitational influence of B_1, the acceleration exerted on B_3 is computed as

$$\ddot{\boldsymbol{R}}_3 = \frac{-\tilde{G}\tilde{M}_1}{\tilde{R}^3_{1,3}}\tilde{\boldsymbol{R}}_{1,3} + \sum_{i=2,i\neq 3}^{N_b} \frac{\tilde{G}\tilde{M}_i}{\tilde{R}^3_{3,i}}\tilde{\boldsymbol{R}}_{3,i} \tag{2.64}$$

and similarly, the acceleration of B_1 given the gravitational influence of B_3 is

$$\ddot{\boldsymbol{R}}_1 = \frac{-\tilde{G}\tilde{M}_3}{\tilde{R}^3_{3,1}}\tilde{\boldsymbol{R}}_{3,1} + \sum_{i=2,i\neq 3}^{N_b} \frac{\tilde{G}\tilde{M}_i}{\tilde{R}^3_{1,i}}\tilde{\boldsymbol{R}}_{1,i} \tag{2.65}$$

Thus, the inertial acceleration of B_3 with respect to B_1 is computed from Equations 2.64 and 2.65 with the assumption that $\tilde{M}_3$ is negligible as

$$\ddot{\boldsymbol{R}}_3 - \ddot{\boldsymbol{R}}_1 = \frac{-\tilde{G}\tilde{M}_1}{\tilde{R}^3_{1,3}}\tilde{\boldsymbol{R}}_{1,3} + \sum_{i=2,i\neq 3}^{N_b} \tilde{G}\tilde{M}_i \left(\frac{\tilde{\boldsymbol{R}}_{3,i}}{\tilde{R}^3_{3,i}} - \frac{\tilde{\boldsymbol{R}}_{1,i}}{\tilde{R}^3_{1,i}}\right) \tag{2.66}$$

Considering any additional accelerations due to higher-order gravity terms, the equation becomes

$$\ddot{\boldsymbol{R}}_{1,3} = \frac{-\tilde{G}\tilde{M}_1}{\tilde{R}^3_{1,3}}\tilde{\boldsymbol{R}}_{1,3} + \sum_{i=2,i\neq 3}^{N_b} \tilde{G}\tilde{M}_i \left(\frac{\tilde{\boldsymbol{R}}_{3,i}}{\tilde{R}^3_{3,i}} - \frac{\tilde{\boldsymbol{R}}_{1,i}}{\tilde{R}^3_{1,i}}\right) + \boldsymbol{a}_i \tag{2.67}$$

where $\boldsymbol{a}_i$ captures the higher-order gravitational accelerations, which in this work are only used to generate lunar orbits.

Depending on the application, the DE440 or DE421 lunar and planetary ephemerides, maintained by NASA's Navigation and Ancillary Information Facility (NAIF) [2, 3], are used to locate each celestial body at each epoch during numerical integration [88, 34]. Additionally, the naif0012 file is used for the accurate conversions between UTC and Ephemeris Time, while the pck00011 file is used to obtain the axis orientation for the inertial frames. Other SPICE kernels, i.e., nep095 and nep097, are included to get the accurate position of planetary satellites with respect to Neptune.

To capture the higher-order gravitational accelerations to which a spacecraft is subject orbiting the Moon, the lunar gravity field is represented by a 100 × 100 degree and order spherical harmonics model. The Moon-fixed frame is defined using principal axes, and the potential function for the deviation of the gravity field from a point mass is defined in this frame as

$$U_L = \frac{GM_L}{r}\left[\sum_{l=2}^{100}\sum_{m=0}^{l}\left(\frac{R_L}{r}\right)^l P_{l,m}\left(\sin(\phi)\right)\left(C_{l,m}\cos(m\lambda) + S_{l,m}\sin(m\lambda)\right)\right] \tag{2.68}$$

where R_L is the reference radius of the Moon, $P_{l,m}$ is the associated Legendre polynomial for degree l and order m, ϕ and λ are the selenocentric latitude and longitude, and $C_{l,m}$ and $S_{l,m}$ are the coefficients of the spherical harmonic expansion [**?**]. In this dissertation, the coefficients of expansion and model parameters are accessed using the 900 × 900 gravity model (GRGM900C), compatible with the DE421 ephemerides [62]. Then, a frame transformation is applied to the vector derivative of the potential function to calculate the acceleration $\boldsymbol{a}_i$ in the Moon-inertial frame.

The solution space of the N-body problem is both complex and chaotic. Consequently, trajectories in this model are often derived by first computing solutions in a lower-fidelity model. For instance, trajectories from the CR3BP can serve as initial guesses, which are then numerically corrected within a point-mass ephemeris model to incorporate the gravitational influence of other bodies and any additional perturbations of the dynamical environment.

2.4 Reference Frame Transformation

The initial conditions for orbit insertion into the Neptunian system used in Chapter 5 are transformed from an inertial frame centered on Neptune to the rotating (synodic) frame used in the NT-CR3BP. The opposite transformation is performed in the same chapter to correct the final optimal trajectories into a point mass ephemeris model, where states are defined in an ICRF centered on Neptune. This transformation is performed as described by Folta et al. in Reference [35].

Given the inertial frame $\mathcal{I}$ centered on B_1, the rotating frame $\mathcal{R}$ defined in the previous section to model the Neptune-Triton CR3BP is obtained considering the position $\tilde{\boldsymbol{R}}^{\boldsymbol{I}}_{1,2}$ and velocity $\tilde{\boldsymbol{V}}^{I}_{1,2}$ of the second primary with respect to the first primary at a specific epoch t_0, which can be obtained via ephemerides. The position and velocity of the spacecraft in the inertial frame are defined via the state vector $\tilde{\boldsymbol{X}}^{I} = [\tilde{\boldsymbol{R}}^{I}_{1,3}, \tilde{\boldsymbol{V}}^{I}_{1,3}]$ with respect to the first primary B_1. With respect to the rotating frame $\mathcal{R}$, the position of the spacecraft is rotated by the angles that define the position of B_2 in $\mathcal{I}$ and translated by the position of the NT-system barycenter. Therefore, the position of the spacecraft in the rotating frame is obtained as

$$\tilde{\boldsymbol{R}}^{R}_{1,3} = \left[{}^{R}\boldsymbol{C}^{I}(E_{t0})\right] \tilde{\boldsymbol{R}}^{I}_{1,3} + \tilde{\boldsymbol{R}}^{R}_{b} \tag{2.69}$$

where $\tilde{\boldsymbol{R}}^{R}_{b}$ is the position of the origin of $\mathcal{I}$ with respect to $\mathcal{R}$ expressed in the rotating frame and $\left[{}^{R}\boldsymbol{C}^{I}(E_{t0})\right]$ is the rotation matrix encoding the rotation of $\mathcal{R}$ with respect to $\mathcal{I}$ about their common origin and is computed as

$$\left[{}^{R}\boldsymbol{C}^{I}(E_{t0})\right] = [\hat{\boldsymbol{x}}_{\boldsymbol{C}}, \hat{\boldsymbol{y}}_{\boldsymbol{C}}, \hat{\boldsymbol{z}}_{\boldsymbol{C}}]^{T} = \left[\frac{\tilde{\boldsymbol{R}}^{I}_{1,2}}{||\tilde{\boldsymbol{R}}^{I}_{1,2}||}, \hat{\boldsymbol{x}}_{\boldsymbol{C}} \times \hat{\boldsymbol{z}}_{\boldsymbol{C}}, \frac{\tilde{\boldsymbol{R}}^{I}_{1,2} \times \tilde{\boldsymbol{V}}^{I}_{1,2}}{||\tilde{\boldsymbol{R}}^{I}_{1,2}, \times \tilde{\boldsymbol{V}}^{I}_{1,2}||}\right]^{T} \tag{2.70}$$

On the other hand, the transformation of the spacecraft's velocity vector from the inertial frame to the synodic frame is obtained considering the time derivative of the rotation matrix $\left[{}^{R}\boldsymbol{C}^{I}(E_{t0})\right]$, which is computed as

$$\left[{}^{R}\dot{\boldsymbol{C}}^{I}(E_{t0})\right] = \left[\frac{d\hat{\boldsymbol{x}}_{\boldsymbol{C}}}{dt}, \frac{d\hat{\boldsymbol{y}}_{\boldsymbol{C}}}{dt}, \frac{d\hat{\boldsymbol{z}}_{\boldsymbol{C}}}{dt}\right]^{T} = \left[\frac{\tilde{\boldsymbol{V}}^{I}_{1,2}}{||\tilde{\boldsymbol{R}}^{I}_{1,2}||} - \hat{\boldsymbol{x}}_{\boldsymbol{C}}\left(\frac{\hat{\boldsymbol{x}}_{\boldsymbol{C}} \cdot \tilde{\boldsymbol{V}}^{I}_{1,2}}{||\tilde{\boldsymbol{R}}^{I}_{1,2}||}\right), \frac{d\hat{\boldsymbol{z}}_{\boldsymbol{C}}}{dt} \times \hat{\boldsymbol{x}}_{\boldsymbol{C}} + \hat{\boldsymbol{z}}_{\boldsymbol{C}} \times \frac{d\hat{\boldsymbol{x}}_{\boldsymbol{C}}}{dt}, 0\right]^{T} \tag{2.71}$$

The third component of the time derivative of the rotation matrix is considered zero, assuming that $\hat{\boldsymbol{z}}$ does not shift significantly relative to the inertial frame in time [35]. With the derivative of the rotation matrix in time, the transformation of the velocity vector is obtained as

$$\tilde{\boldsymbol{V}}^{R}_{1,3} = \left[{}^{R}\boldsymbol{C}^{I}(E_{t0})\right] \tilde{\boldsymbol{V}}^{I}_{1,3} + \left[{}^{R}\dot{\boldsymbol{C}}^{I}(E_{t0})\right] \tilde{\boldsymbol{R}}^{I}_{1,3} + \tilde{\boldsymbol{V}}^{R}_{b} \tag{2.72}$$

where $\tilde{\boldsymbol{V}}_b^R$ is the velocity of the origin of $\mathcal{I}$ with respect to $\mathcal{R}$ expressed in the rotating frame. Since the rotation matrix $\left[{}^R\boldsymbol{C}^I(E_{t0})\right]$ is orthogonal, the rotation from the rotating synodic frame to the inertial frame is computed as

$$\begin{aligned}
\tilde{\boldsymbol{R}}_{1,3}^I &= \left[{}^R\boldsymbol{C}^I(E_{t0})\right]^T \tilde{\boldsymbol{R}}_{1,3}^R + \tilde{\boldsymbol{R}}_b^I \\
\tilde{\boldsymbol{V}}_{1,3}^I &= -\left[{}^R\boldsymbol{C}^I(E_{t0})\right]^T \left[{}^R\dot{\boldsymbol{C}}^I(E_{t0})\right] \left[{}^R\boldsymbol{C}^I(E_{t0})\right]^T \tilde{\boldsymbol{R}}_{1,3}^R + \left[{}^R\boldsymbol{C}^I(E_{t0})\right]^T \tilde{\boldsymbol{V}}_{1,3}^R + \tilde{\boldsymbol{V}}_b^I
\end{aligned} \tag{2.73}$$

The transformation from the rotating synodic frame to the inertial frame can also be rewritten in the same form used in Equations 2.69 and 2.72 considering the rotation matrix $\left[{}^I\boldsymbol{C}^R(E_{t0})\right]^T$ obtained as the transpose of $\left[{}^R\boldsymbol{C}^I(E_{t0})\right]$ [35]. The states obtained after the transformation in Equations 2.69 and 2.72 can be normalized using three characteristic quantities l^*, m^*, and t^* and propagated using the equations of motion in 2.13, while the state in the inertial frame can be propagated considering the point mass gravity of N_b-bodies in an ephemeris model described with the Equation 2.67.

Chapter 3

Clustering Techniques

Clustering is a method for grouping members of a dataset based on a specified set of features without requiring a human-in-the-loop. Clustering algorithms are often categorized as hierarchical-based, partitioning-based, density-based, grid-based, and/or model-based [47]. Furthermore, hard clustering uniquely assigns each member of a dataset to a single cluster, whereas fuzzy clustering assigns probabilities of membership to each cluster. In astrodynamics, clustering has been used to detect bounded motions around small bodies [108] and near distant retrograde orbits [79], group periodic orbits that are independently computed [46], extract motion primitive sets that summarize families of trajectories [100], and summarize a wide variety of trajectories in the circular restricted three-body problem and ephemeris model at a single or multiple energy levels [9, 7, 10, 12].

3.1 DBSCAN

The Density Based Spatial Clustering of Applications with Noise [32] is used in the process to identify clusters or trajectory arcs with similar geometries, from which motion primitives are extracted. This algorithm is agnostic to the data's shape and the number of groups, making it particularly useful for clustering large datasets with non-uniform groups of data. This algorithm is governed by two parameters: ϵ, which is the size of the neighborhood around a point S_i of the dataset, and n_{sample} is the number of samples in a neighborhood for a point to be considered as a core point [32].

In DBSCAN, the output is a vector indicating the cluster number, or label, to which each point has been assigned. Labels are assigned only to core points and boundary points. A core point is a point of the dataset that has at least n_{sample} points in its neighborhood. Points that are not core points but lie in the neighborhood of another point are defined as border points. Points that are neither core points nor border points are defined as noise points and are indicated with a zero entry in the label vector.

To understand the algorithm steps, it is also important to define the concepts of directly density-reachable, density-reachable, and density-connected points. In particular, when a point S_i is in the neighborhood of a point S_j and S_j is also a core point, then S_i is defined as directly density-reachable from S_j. Otherwise, S_i is density-reachable from S_j if there exists a chain of points $S_1, s_2, ..., s_n$ where $S_j = s_1$ and $S_i = s_n$, and each point S_{a+1} is directly density-reachable from S_a. Finally, two points S_i and S_j are density-connected if there is a point S_k such that both S_i and S_j are density-reachable from S_k [32].

DBSCAN clusters points in a dataset through five steps. First, the algorithm is initialized on a point S_i and marks S_j as visited. Then all the neighbors of S_i are retrieved, i.e., all the points lying within a radius ϵ from S_i. If the neighbors are larger or equal to the values of n_{sample}, then S_i is defined as a core point, and a new cluster is created. The cluster is iteratively expanded by visiting all the points that are density-reachable from S_i. In this way, a cluster includes core points and border points. However, if a point S_i does not have n_{sample} points in its neighborhood and is not in any other point's neighborhood, then it is marked as noise. This definition could be changed later in the process if S_i is within the neighborhood of another point. In either case, no points are density-reachable from S_i, and DBSCAN visits the next point in the database. The process is repeated for every point in the dataset until all of them have been marked as visited [32].

In the cluster expansion step, DBSCAN visits all the neighbors of S_i. Then, if a neighbor S_j is also a core point, it adds its neighbors to the list of points to be visited (expanding the cluster). The process stops when all the points in the list have been checked and no more points can be added to the list [32].

3.2 HDBSCAN

The Hierarchical Density-Based Spatial Clustering of Applications with Noise is a clustering algorithm developed by Campello, Moulavi, and Sander [18]. HDBSCAN is a hard clustering algorithm that discovers clusters of arbitrary shape and density that are separated by an unknown or nonconstant distance. Similarly to DBSCAN, HDBSCAN does not require a priori knowledge of the number of clusters and labels data in insufficiently sampled regions as noise. Because of these characteristics, HDBSCAN has successfully been used in a variety of fields, from medicine [40] to computer vision [63]. This clustering algorithm has also previously been demonstrated to successfully cluster spacecraft trajectories in a chaotic dynamical model [9, 7]. In this work, HDBSCAN is leveraged in the identification of trajectories with a similar perilune evolution, as it will be described in Chapter 4, and in the creation of geometrically distinct trajectory arcs from natural solutions of the NT-CR3BP, as will be discussed in Chapter 6.

HDBSCAN uses a hierarchical and density-based approach to group the N members of a dataset, each described by m-dimensional feature vectors that capture user-specified characteristics of interest. The governing hyperparameters n_{sample} and n_{size} are, respectively, the number of points in a neighborhood for a point to be considered a core point and the minimum number of points to form a cluster. Any members that are not assigned to a group are labeled as noise. Based on a modification presented by Malzer and Baum, a quantity ϵ_{merge} can be used to specify a minimum threshold on when members are split into multiple clusters.

The algorithm transforms the dataset into a hierarchical structure following six steps. First, the core distance d_{core} is computed for each member. For the ith member, this quantity is defined as the distance of the feature vector $\boldsymbol{\nu}_i$ from its $(n_{sample} - 1)$th nearest neighbor, assessed using a specified distance measure. The core distance is used to calculate the mutual reachability distance (MRD) between each pair of feature vectors $\boldsymbol{\nu}_i$ and $\boldsymbol{\nu}_j$ as $d_{mreach} = \max(d_{core}(\boldsymbol{\nu}_i), d_{core}(\boldsymbol{\nu}_j), d(\boldsymbol{\nu}_i, \boldsymbol{\nu}_j))$, where $d(\boldsymbol{\nu}_i, \boldsymbol{\nu}_j)$ is the distance between $\boldsymbol{\nu}_i$ and $\boldsymbol{\nu}_j$. Using this information, a mutual reachability graph is constructed with N nodes defined as the feature vectors and the edges between each pair of nodes weighted by the MRD of the associated feature vectors. This graph is summarized using a minimum spanning tree (MST) that is augmented by adding self-loops to each node that are weighted by the core distance of its feature vector.

HDBSCAN then constructs a dendrogram from this augmented MST to produce a clustering hierarchy. Clusters are identified from this hierarchy as groups of members that are sufficiently stable as a function of density and possess a size that is above a minimum value, denoted n_{size}. Then clusters with a distance below the threshold ϵ_{merge} are merged. If the local neighborhood of a member of the dataset does not encompass at least $(n_{sample} - 1)$ neighbors, that member is labeled as a noise point. The output of this process is a set of labels assigning each of the N members of the dataset to either a cluster or a noise group. In this paper, HDBSCAN is accessed via the *hdbscan* clustering library in Python [70].

The Density-Based Clustering Validation (DBCV) index introduced by Moulavi et al. is used to assess the quality of a clustering result generated by HDBSCAN [76]. The DBCV index measures the ratio of the inter-cluster separation to the intra-cluster density with values between -1 and 1; a high value of the DBCV index indicates a better clustering result with clusters that are more tightly packed and well-separated. Mathematically, the DBCV index is defined for a dataset that has been grouped into n_{clust} clusters as

$$DBCV = \sum_{i=1}^{n_{clust}} \frac{C_i}{n_{noise}} V_C(C_i) \tag{3.1}$$

where n_{noise} is the number of noise points and $V_C(C_i)$ is the validity index of cluster C_i, defined as

$$V_C(C_i) = \frac{\min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j)) - DSC(C_i)}{\max(\min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j)), DSC(C_i))} \tag{3.2}$$

where $DSPC$ is the density separation of a pair of clusters, defined as the minimum reachability distance between the internal nodes of the MST of clusters C_i and C_j; and DSC is the density sparseness of a cluster, defined as the maximum edge weight of the internal edges in the MST of the cluster C_i. This validity index compares the internal density compactness of a cluster and the density separation between two clusters, with a positive value of $V_C(C_i)$ indicating that a better cluster is one that is compact and well separated from other clusters.

Chapter 4

Clustering Approach to Identify Long-Term Stable Orbits about Planetary Moons

This chapter presents a clustering-based approach to summarizing a wide variety of trajectories that are numerically generated in a high-fidelity lunar gravity model. This summary is used to extract insight into the solution space and locate motions with a bounded evolution of perilune that may supply candidates for low lunar frozen and quasi-frozen orbits. First, initial conditions are defined as perilunes with distinct combinations of orbital elements. To support a proof of concept, these perilunes possess a fixed semi-major axis of 1838 km and an initial epoch on January 1, 2025. Trajectories are generated for up to 180 days from these initial conditions in a 100 × 100 lunar gravity model with the point mass gravity of the Earth and Sun. This model fidelity is selected to balance prediction accuracy against computational time. Consistent with previous analyses of frozen orbits, each trajectory is characterized by the time evolution of the eccentricity e and argument of periapsis ω at each perilune in a Moon-fixed frame defined using principal axes. The evolution of perilune over 180 days is then summarized to produce a finite-dimensional feature vector that encodes its size and shape. Next, HDBSCAN is used to cluster these feature vectors in a two-step process [17]. Each cluster corresponds to trajectories with a geometrically similar evolution of perilune. The result is a clustering-based summary of the geometries exhibited by trajectories near the Moon that is also used to identify candidates for low lunar frozen or quasi-frozen orbits. The clustering-based framework for summarizing lunar trajectories and extracting bounded orbits consists of the following steps:

(1) Numerically generating a set of trajectories in a selected dynamical model

(2) Describing each trajectory using a finite-dimensional feature vector

(3) Clustering the dataset to produce groups of spatiotemporally similar trajectories

(4) Summarizing each cluster using a representative member

(5) Merging clusters with geometrically similar representatives

4.1 Step 1: Generating a Set of Trajectories

To generate a set of lunar trajectories, a wide range of initial conditions is defined using Keplerian orbital elements at a fixed epoch and semi-major axis. Although epoch and semi-major axis are both variables that impact the characteristics of the associated trajectories, they are constrained in this work to reduce the size of the dataset and support a proof of concept. The ranges and step sizes of each orbital element are listed in Table 4.1. All the possible combinations of these orbital elements result in 58,608 initial conditions that lie above the lunar radius, assumed to equal 1738 km in this work. Each initial condition is propagated for up to 180 days or until it reaches the lunar radius. This numerical propagation is performed in the high-fidelity dynamical model presented in Section 2.3. To limit data storage requirements, only the perilune states are recorded along each trajectory. These trajectories were generated by Dr. Natasha Bosanac.

Table 4.1: Ranges of orbital elements used to define initial conditions

Orbital Element	Range	Step
Epoch, t_0	January 1, 2025 00:00.000 UTC	-
Semi-major axis, a	1838 km	-
Eccentricity, e	$[10^{-4}, 0.1]$	0.005
Inclination, i (in Moon-fixed frame)	$[0.001°, 179.999°]$	5°
Right ascension of the ascending node (RAAN), Ω (in Moon-fixed frame)	$[0°, 360°]$	30°
Argument of periapsis (AOP), ω (in Moon-fixed frame)	$[0°, 360°]$	30°

4.2 Step 2: Describing Each Trajectory via a Feature Vector

Consistent with traditional analyses of frozen orbits, each lunar trajectory in this work is represented by the evolution of perilune over time. Researchers have commonly searched for frozen orbits as trajectories with a bounded variation in the eccentricity and argument of periapsis [15, 94, 37]; some researchers also use alternative orbital element sets [89, 29]. In this dissertation, each perilune is partially described using the variables $p = e\cos(\omega)$ and $q = e\sin(\omega)$ because they possess equivalent value ranges but visualized in an $e - \omega$ polar plot for clarity. To demonstrate the value of this low-dimensional representation, Figure 4.1a displays a 4-day segment of a lunar trajectory in the Moon-fixed frame that is generated from an initial perilune with the following orbital elements: $e = 0.025,\ i = 85°,\ \omega = 0°,\ \Omega = 210°$. The associated evolution of each perilune over 180 days in the Moon-fixed frame is then plotted in black in Figure 4.1b on an $e - \omega$ polar plot with the first and last perilune located by red and blue markers, respectively. Using this figure as an example, trajectories that are generated for at least 27.3 days tend to possess an evolution of perilune that exhibits multiple revolutions in the pq-plane and $e - \omega$ polar plot, with each revolution occurring over one lunar rotational period, i.e., 27.3 days [59]. Later steps in the clustering framework exploit this geometric property.

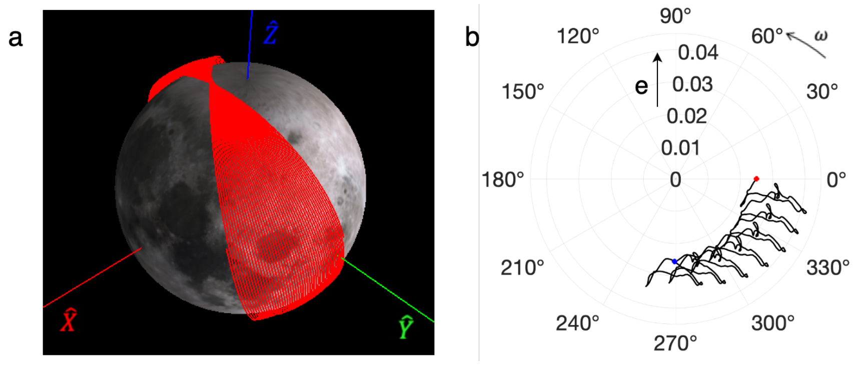


Figure 4.1: a) A 4-day segment of a lunar trajectory in the Moon-fixed frame generated from a perilune with $e = 0.025,\ i = 85°,\ \omega = 0°,\ \Omega = 210°$ and b) Associated evolution of perilune over 180 days in an $e - \omega$ polar plot; red and blue markers indicate the initial and final perilunes

The evolution of the perilunes along each trajectory is summarized using a feature vector that is input to a clustering algorithm. In general, the feature vector $\boldsymbol{\nu}_{gp,i}$ is a finite-dimensional representation of the characteristics of interest for the ith member of a dataset $\mathcal{D}$. In this work, this feature vector is constructed to capture the spatiotemporal evolution of these perilunes; thus, it is labeled with the subscript gp, indicating the description in geometry and phasing. First, a fixed number of m_p perilunes are evenly sampled from the perilunes along each trajectory. Using a fixed number of samples for all trajectories enables the definition of a feature vector with a fixed length along the entire dataset and, therefore, the use of fast distance measures during clustering. In this work, $m_p = 90$ is selected empirically: for the trajectories in our dataset, 15 perilunes along each lunar rotational period can sufficiently describe the perilune evolution shape. Then, $\boldsymbol{s}_{i,k}$ is defined as a unit vector directed from the kth perilune to the $k+1$th perilune along the ith trajectory in the pq plane. Mathematically, this unit vector is defined as

$$\boldsymbol{s}_{i,k} = \frac{[p_{i,k+1} - p_{i,k}, q_{i,k+1} - q_{i,k}]}{||[p_{i,k+1} - p_{i,k}, q_{i,k+1} - q_{i,k}]||} \tag{4.1}$$

At the kth perilune along the ith trajectory generated over an integration time of t_{max}, a normalized elapsed time is also defined as

$$t'_{i,k} = \frac{t_{i,k}}{t_{max}} \tag{4.2}$$

Using these two quantities, the feature vector for the ith trajectory is defined as

$$\boldsymbol{\nu}_{gp,i} = [\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, ..., \boldsymbol{s}_{i,m-1}, t'_{i,1}, t'_{i,2}, ...t'_{i,m}] \tag{4.3}$$

producing a $(3m-2)$-dimensional description. This feature vector approximates the shape of the discrete path formed by the perilunes along a trajectory in the pq-plane as well as its phasing.

4.3 Step 3: Clustering the Trajectories by Geometry and Phasing

To cluster the feature vectors generated in Step 2 using HDBSCAN, multiple governing parameters must be selected. First, the Euclidean distance is used to assess the difference between feature vectors $\boldsymbol{\nu}_{gp,i}$ and $\boldsymbol{\nu}_{gp,j}$, computed as

$$d_e(\boldsymbol{\nu}_{gp,i}, \boldsymbol{\nu}_{gp,j}) = \sqrt{\sum_{k=1}^{m_p} (\boldsymbol{s}_{i,k} - \boldsymbol{s}_{j,k})^T (\boldsymbol{s}_{i,k} - \boldsymbol{s}_{j,k}) + \sum_{k=1}^{m_p} (t'_{i,k} - t'_{j,k})^2} \tag{4.4}$$

Although the Euclidean distance only compares two time-ordered sequences as opposed to two geometric paths, this distance metric is used because it enables fast and computationally tractable clustering for a large dataset. Then, a grid search is used along with cluster validation techniques to select suitable values of n_{sample} and n_{size}. This grid search is performed by generating clustering results for various combinations of $n_{sample} = [2, 26]$ with a step size of 4 and $n_{size} = [30, 100]$ with a step size of 10. After clustering the generated set of trajectories using HDBSCAN for all possible combinations of these parameters, n_{sample} and n_{size} are selected to balance producing a high DBCV index with identifying a reasonable number of clusters and noise points. With these goals, $n_{sample} = 2$, $n_{size} = 50$ and $\epsilon_{merge} = 0.0$ are selected to produce $DBCV = 0.1745$, $n_{noise} = 30\%$ of the dataset, and $n_{clust} = 402$ clusters; this set of clusters is labeled $\mathcal{C}_{gp}$ to reflect that the trajectories are grouped according to both geometry and phasing. The percentage of trajectories designated as noise is relatively high; however, ongoing work has focused on addressing this issue.

Across the 402 clusters in $\mathcal{C}_{gp}$, trajectories with perilune evolutions that exhibit similar geometry and phasing are grouped together. As an example, Figure 4.2 displays the perilune evolution of a subset of the trajectories in four clusters in the $e - \omega$ polar plot; each path is uniquely colored. The black curve highlights the perilune evolution of a single trajectory to facilitate comparison, while red and blue markers locate the first and last perilunes, respectively. Within each subfigure, the evolution of the perilunes along each trajectory is geometrically similar, with a similar phasing. However, the location and secular drift in the $e - \omega$ polar plot vary across the cluster. In addition, Figures 4.2a and 4.2b display two clusters capturing perilune evolutions with a similar geometry but distinct phasing. Alternatively, Figures 4.2c and 4.2d display two clusters that, when compared, contain geometrically distinct paths traced out by the perilunes in the $e - \omega$ polar plot. Accordingly, these four clusters demonstrate the capability of the clustering framework to group the generated lunar trajectories based on the geometry and phasing of their evolution of perilune.

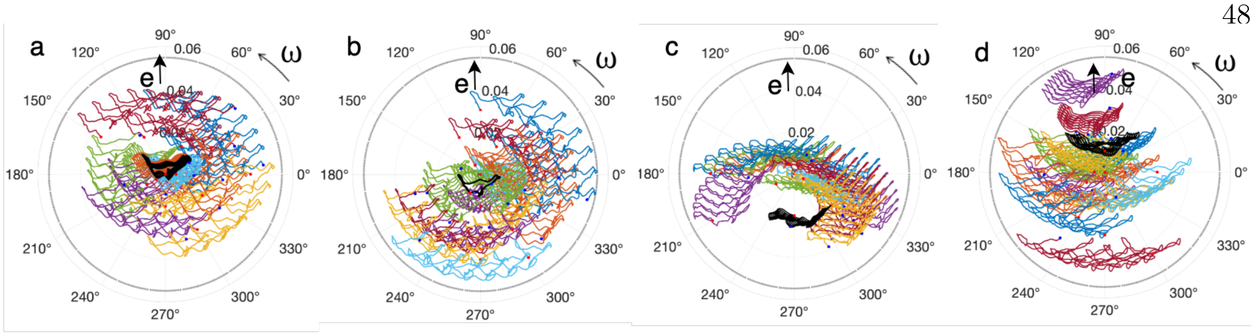


Figure 4.2: Perilune evolution of selected members of four clusters: sample member in black with red and blue points indicating the initial and final perilunes, respectively

4.4 Step 4: Extracting a Cluster Representative

To summarize each cluster, a cluster representative is extracted as the trajectory with the most bounded evolution of perilune in the pq-plane. This trajectory is identified as the sequence of perilunes that possesses the smallest cumulative distance, d_{seq}, between revolutions in the pq-plane. Recall that each revolution traced out by the perilunes in the pq-plane occurs over one lunar rotational period. Accordingly, a perilune path completes $w = T_p/T_l$ revolutions in the pq-plane, where T_p is the integration time along the trajectory and T_l is the lunar rotational period. As a result, there are approximately $\tau = \lfloor m_p/w \rfloor$ perilunes sampled along each revolution. Using this approximation, the distance is calculated between the ith sampled perilune that lies along the jth revolution and the $i+\tau$th sampled perilune in the pq-plane. This distance is displayed conceptually in Figure 4.3 as $d([p,q]_i, [p,q]_{i+\tau})$. Along the jth revolution, the Euclidean distances between each sampled perilune and the associated perilunes along the $j+1$th revolution are averaged. These distances are computed between all subsequent revolutions and summed to produce the cumulative distance d_{seq}. The member of a cluster with the smallest value of d_{seq} is selected as the representative trajectory of the cluster.

Considering a state along the perilune trajectory $\boldsymbol{x}_i$, where i is the index representing the ith cycle of the trajectory, the distance between subsequent revolutions is computed as the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+\tau}$, where τ is defined above as the number of sample along each perilune.

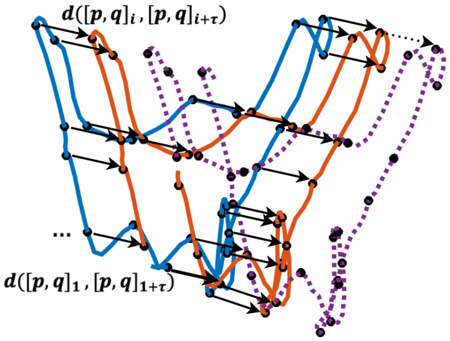


Figure 4.3: Calculating the distance between perilunes along neighboring revolutions in the pq-plane to identify a representative member of a cluster

This process is repeated on a subsample of perilunes along each revolution, and then the distances obtained are used to compute the average distance between each revolution. A representative trajectory is extracted from each cluster in $\mathcal{C}_{gp}$ to support further analysis. This is the member of the cluster with the smallest average distance between each revolution.

Figure 4.4 displays an example of a cluster representative. In Figure 4.4a, the evolution of perilune is shown in the Moon-fixed frame with the red and blue markers indicating the first and last perilunes, respectively. The evolution of perilune for this representative trajectory is also depicted in Figure 4.4b in the $e-\omega$ polar plot. In this particular case, the evolution of perilune associated with this cluster representative exhibits only a slight drift between subsequent revolutions, both in the pq-plane and in the Moon-fixed frame. For additional examples, the black highlighted paths in Figure 4.2 are the representatives of those four clusters.

4.5 Step 5: Merging Clusters with Geometrically Similar Representatives

Members of multiple clusters in $\mathcal{C}_{gp}$ may exhibit a geometrically similar evolution of perilune but are correctly separated due to distinct phasing of the initial and final perilunes. Accordingly, a second clustering step is used to merge clusters with a similar geometry, independent of phasing. This additional step is completed by using HDBSCAN to cluster only the representatives of each cluster from $\mathcal{C}_{gp}$ with a new geometric feature vector and distance measure; the resulting set of

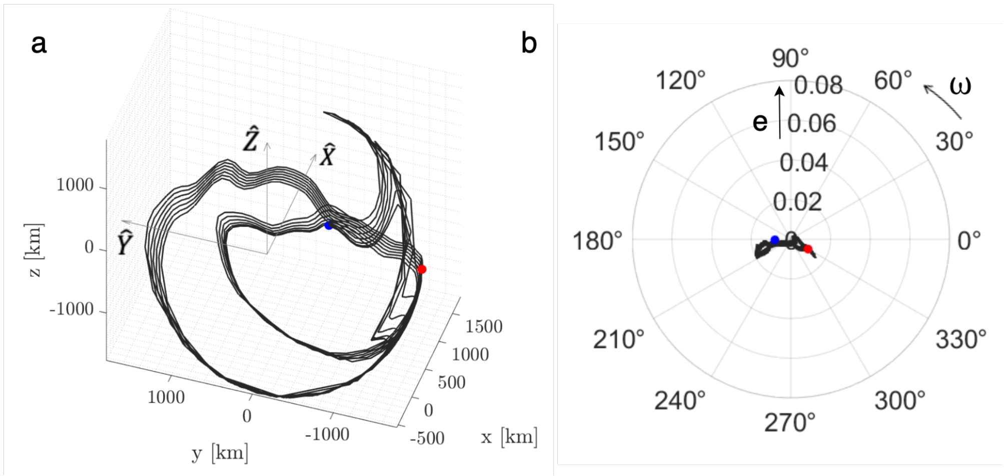


Figure 4.4: Perilune path associated with a cluster representative in the a) Moon-fixed frame and b) $e - \omega$ polar plot

clusters of representatives is labeled $\mathcal{C}_r$. For each group of representatives in $\mathcal{C}_r$, their associated clusters from $\mathcal{C}_{gp}$ are merged. By applying this second clustering step to the cluster representatives, the increased computational expense associated with this new feature vector and distance measure does not become burdensome. The result of this second step is a set of clusters, each containing trajectories with a similar geometry regardless of the relative phasing of the initial and final perilunes.

A geometric feature vector is defined to describe the boundary of the path traced out by the perilunes along each cluster representative in the pq-plane. This boundary is computed using shape-related functions in MATLAB [68]. First, the `alphaShape` function is applied to the ordered set of perilunes along the cluster representative to construct a polygon shape in the pq coordinates. The polygon obtained with this function entirely encloses the perilune path, creating a convex shape with no internal holes. Then, the pq-coordinates of the perilunes that lie at the boundary of this polygon shape are extracted using the `boundaryFacets` function. Linear interpolation is used to extract a fixed number of pq coordinates that lie along each boundary to ensure that a boundary with the same resolution describes all cluster representatives; the number of points is selected as the largest number of boundary perilunes calculated by the `boundaryFacets` function

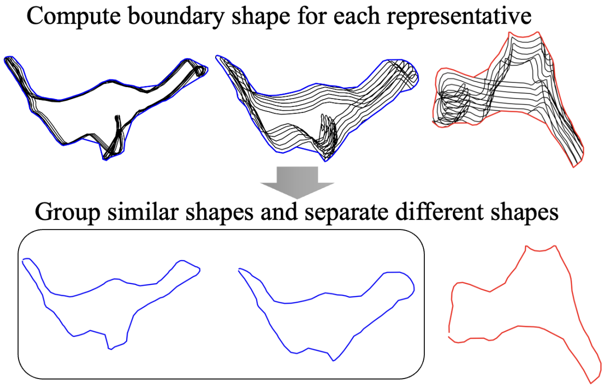

Compute boundary shape for each representative
Group similar shapes and separate different shapes

Figure 4.5: Computed boundaries of the paths traced out by perilunes along three trajectories in the $e - \omega$ plane

along the entire set of representatives. Then, the geometric feature vector $\boldsymbol{\nu}_{g,i}$ describing the ith cluster representative is defined as

$$\boldsymbol{\nu}_{g,i} = [p_{b_1}, q_{b_1}, ..., p_{b_k}, q_{b_k}] \tag{4.5}$$

using the interpolated set of k boundary points b_j for $j = [1, k]$. Examples of the boundaries obtained from three sample perilune paths (black) in the pq-plane are displayed in Figure 4.5. In this figure, the left and center cluster representatives with blue boundaries possess a similar geometry and are, therefore, grouped together after clustering. The rightmost cluster representative with the red boundary, however, possesses a distinct geometry from the other two representatives.

To cluster the geometric feature vectors describing the representatives of $\mathcal{C}_{gp}$ via HDBSCAN, multiple governing parameters must be selected. First, the values of the hyperparameters governing HDBSCAN are modified to $n_{sample} = 1$, $n_{size} = 2$, and $\epsilon_{merge} = 0.08$ based on visual inspection of the clusters to accommodate the smaller dataset. The choice of $n_{sample} = 1$ and $n_{size} = 2$ reduces the algorithm to a single linkage. In fact, the value $n_{size} = 2$ disables the pruning of the MST branches, i.e., of the hierarchical representation of the dataset, similarly to a robust single linkage [17]. Additionally, by setting $n_{sample} = 1$, the core distance of each point is reduced to zero, and the MRD becomes the Euclidean distance between two points as in the ordinary single linkage approach [70]. Next, the modified Hausdorff distance d_{mhd} is used as a distance measure during this second clustering step to capture geometric differences in each path, independent of phasing. This distance measure is mathematically defined as:

$$d_{mhd}(\boldsymbol{\nu}_{g,i}, \boldsymbol{\nu}_{g,j}) = d_{mhd,f}(\boldsymbol{\nu}_{g,i}, \boldsymbol{\nu}_{g,j}) + d_{mhd,b}(\boldsymbol{\nu}_{g,j}, \boldsymbol{\nu}_{g,i}) \tag{4.6}$$

where

$$d_{mhd,f}(\boldsymbol{\nu}_{g,i}, \boldsymbol{\nu}_{g,j}) = \max_{i=1,..m}(\min_{j=1,..m} \|\boldsymbol{\nu}_{g,i} - \boldsymbol{\nu}_{g,j}\|_2) \tag{4.7}$$

$$d_{mhd,b}(\boldsymbol{\nu}_{g,i}, \boldsymbol{\nu}_{g,j}) = \max_{j=1,..m}(\min_{i=1,..m} \|\boldsymbol{\nu}_{g,i} - \boldsymbol{\nu}_{g,j}\|_2) \tag{4.8}$$

Due to the complexity of computing this distance measure, a higher computational time is required when compared to the Euclidean distance. However, the size of the reduced dataset of cluster representatives makes this computational time reasonable.

HDBSCAN is used to cluster the geometric feature vectors describing only the representatives of the clusters in $\mathcal{C}_{gp}$. This second clustering step produces 41 clusters of representatives with similar geometry and 55 noise points. Representatives that exist in the same cluster in $\mathcal{C}_r$ indicate that their associated clusters from $\mathcal{C}_{gp}$ should be merged. The cluster representatives labeled as noise points in $\mathcal{C}_r$, however, indicate that 55 of the original clusters from $\mathcal{C}_{gp}$ should not be merged. Following the cluster merging process, there are 96 clusters of trajectories, labeled as the global clustering result $\mathcal{C}_g$ throughout the remainder of the work to reflect that the trajectories are only grouped by geometry. Polar plots of the representative trajectories of these 96 clusters appear in the Appendix.

Visual inspection reveals that the merging process successfully combines trajectories with a similar geometry, regardless of phasing. An example of a merged cluster in $\mathcal{C}_g$ is displayed in Figure 4.6a, with each of 12 representatives from the original clusters in $\mathcal{C}_{gp}$ uniquely colored. Each cluster representative from $\mathcal{C}_{gp}$ exhibits a similar geometry in the $e-\omega$ plane. Figure 4.6b also includes selected members of all 12 clusters from $\mathcal{C}_{gp}$, displayed with the same color as the representative but a thin, transparent curve. The gray circle represents the value of eccentricity at impact with the Moon's surface when $a = 1838$ km. The region of the $e-\omega$ plane encompassed by members of this larger merged cluster indicates the region of existence of lunar trajectories with a similar geometric evolution in perilune but varying phasing, drift, and average eccentricities.

In addition to exhibiting a similar geometry in the $e - \omega$ polar plot, the 12 representatives that are plotted in Figure 4.6a also exhibit a similar evolution of the remaining orbital elements. Figure 4.7 displays the range of values of the altitude, argument of perilune, eccentricity, inclination, and RAAN of perilunes along each cluster representative; each angle is calculated in the Moon-fixed frame. The orbital elements of each representative are colored using the same color scheme as Figure 4.6. The 12 representatives all begin with perilunes that exist at an inclination of $i = 75°$ in the Moon-fixed frame and exhibit only a small variation in this angle over time. Across all 12 representatives, the perilune altitude also varies by approximately 30 km over 180 days.

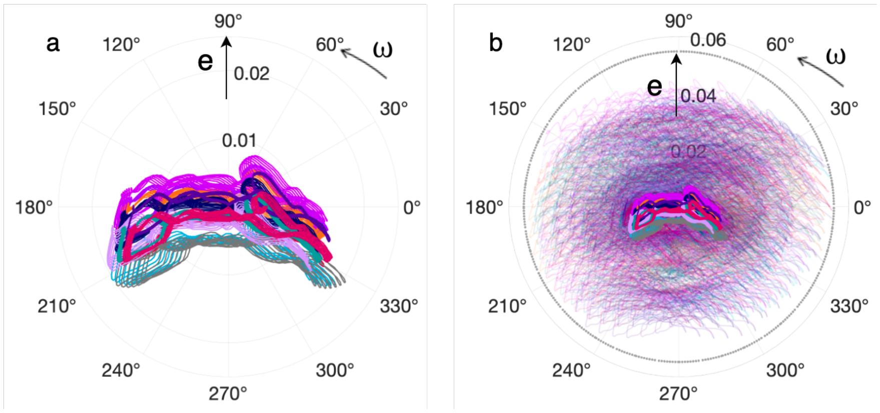


Figure 4.6: a) 12 representatives of clusters in $\mathcal{C}_{gp}$ that are grouped based on geometry in the second clustering step and b) Selected members of the merged cluster in $\mathcal{C}_g$

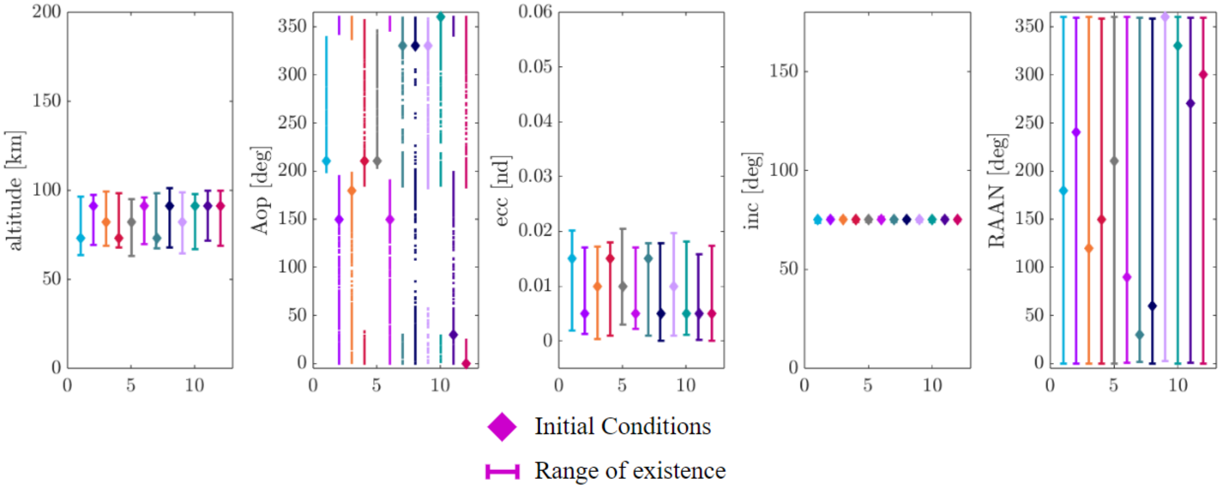


Figure 4.7: Initial orbital elements (diamond) with angles in the Moon-fixed frame associated with the 12 cluster representatives in Figure 4.6a and their ranges of values during propagation

Chapter 5

Application of Clustering Approach Identifying Low Lunar Long-Lifetime Orbits in a High-Fidelity Model

Using the approach presented in Chapter 4, this chapter presents a broader analysis of the clustering-based summary of lunar trajectories generated in a high-fidelity model. First, several candidates for lunar frozen or quasi-frozen orbits are identified across the entire clustering result. Note, the phrase 'candidate for a frozen or quasi-frozen orbit' is used in this work because 1) trajectories are only generated for 180 days, and 2) the analysis is initially performed on the evolution of the eccentricity and argument of perilune; subsequent analysis requires examining the evolution of all orbital elements over longer time intervals. In addition, the clusters of trajectories with a geometrically similar evolution of perilune are used to identify local trends in the orbital elements that lead to changes in the orbit lifetime.

5.1 Identifying Candidates for Frozen Orbits

By analyzing the final clustering result $\mathcal{C}_g$ from Chapter 4, trajectories with a tightly bounded evolution of perilune in the pq-plane and a lifetime of 6 months are analyzed as candidates for frozen orbits. For example, consider the merged cluster of trajectories with a perilune evolution, as displayed in Figure 4.6. Figure 5.1 displays the perilunes along these 12 representative trajectories in Cartesian coordinates in the Moon-fixed frame using a different color scheme; each subfigure displays an alternative three-dimensional view for clarity. This figure reveals that the 12 representative trajectories can be divided into two groups. The perilune paths that are colored in shades of blue

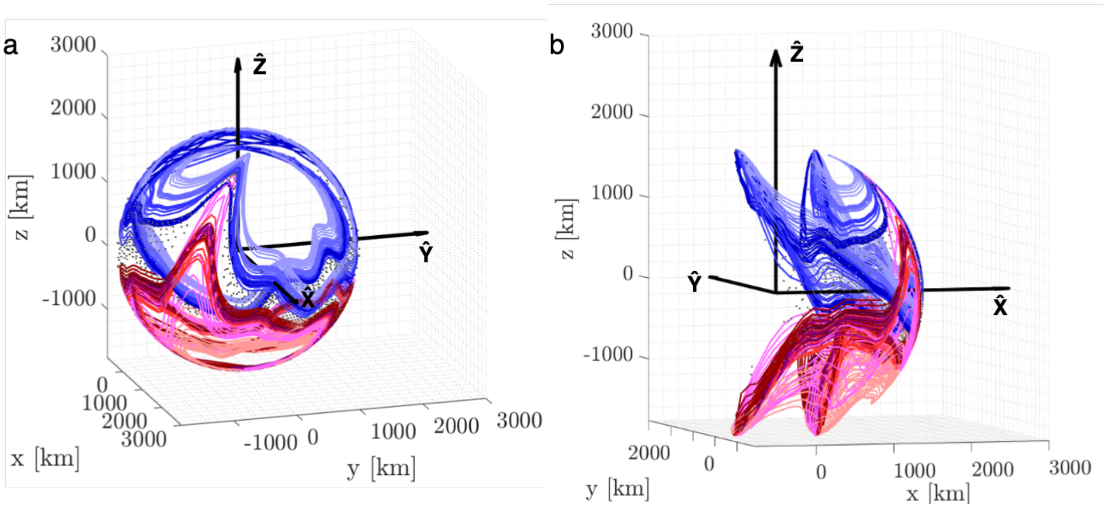


Figure 5.1: Three-dimensional views of the uniquely-colored paths traced out by perilunes of 12 grouped cluster representatives in two different orientations in the Moon-fixed frame

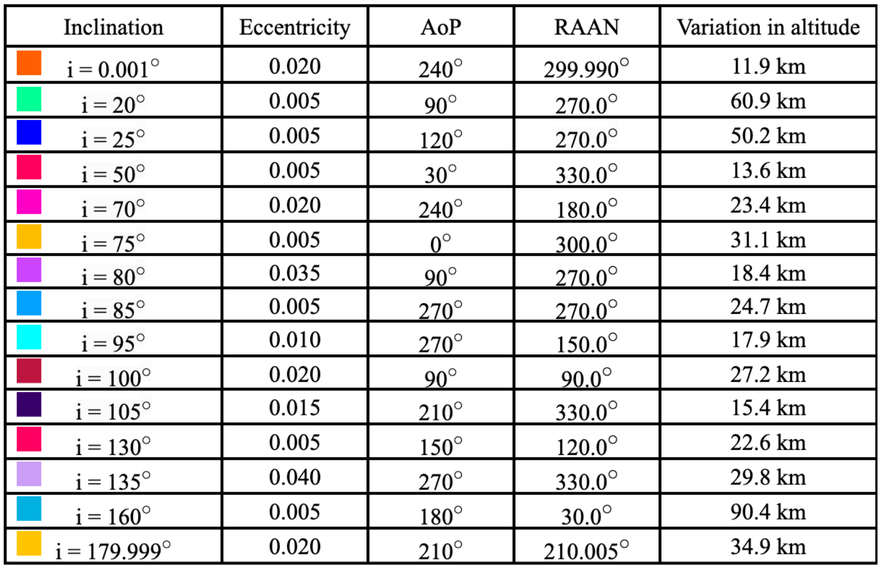
revolve around the $+\hat{\boldsymbol{Z}}$-axis of the Moon-fixed frame, aligned with the third lunar principal axis, twice every lunar rotational period. In the polar plot, the associated values of the argument of perilune in the Moon-fixed frame lie predominantly in the range $[0°, 180°]$. Similarly, the perilune paths that are colored in shades of red perform two revolutions around the $-\hat{\boldsymbol{Z}}$-axis of the Moon-fixed frame and possess arguments of perilune that are predominantly in the range $[180°, 360°]$. The perilunes along all 12 representative trajectories also predominantly pass over the $+\hat{\boldsymbol{X}}$ hemisphere of the Moon, corresponding to the first lunar principal axis and mean direction to the Earth.

The evolution of the perilune along grouped cluster representatives supports the visual identification of one or more candidates for low-lunar, frozen, or quasi-frozen orbits. To understand this process, consider an analogy to a stable periodic orbit with nearby quasi-periodic orbits in the well-known planar circular restricted three-body problem. In this dynamical model, Poincaré maps are frequently employed to analyze the structure of the solution space. On a suitably constructed Poincaré map, stable periodic orbits appear as a fixed point that is surrounded by concentric closed curves corresponding to quasi-periodic orbits. When generating these Poincaré maps, a stable periodic orbit is rarely computed exactly. However, the presence of concentric curves and, therefore, a family of quasi-periodic orbits indicates the existence of the associated stable periodic orbit, supporting the localization of its precise trajectory with the aid of differential corrections or other numerical methods [90].

Although the ephemeris model used in this work is not autonomous and does not admit periodic orbits, the presented analogy is still useful. In Figures 4.6 and 5.1, the representative trajectories exhibit various levels of drift in the perilunes during each subsequent lunar rotational period. In this figure, the lower the drift, the darker the shade of blue or red of the path traced out by the perilunes of the associated trajectory. In this case, the perilune paths that are colored in the darkest shades of blue and red correspond to two trajectories with a perilune evolution that exhibits a low drift in the eccentricity over 180 days: one trajectory with a perilune predominantly over the northern hemisphere and the other over the southern hemisphere, supplying two suitable candidates for lunar frozen or quasi-frozen orbits.

The manual identification of candidates for low lunar frozen and quasi-frozen orbits is repeated across all clusters in $\mathcal{C}_g$. For initial conditions that are constrained to possess a semi-major axis of 1838 km at January 1, 2025, 00:00.000 UTC with the discretization scheme outlined in Step 1, a total of 15 candidates have been identified. These candidates possess various geometries in the evolution of perilune in both the *pq*-plane and the Moon-fixed frame. The initial conditions used to generate these 15 trajectories are displayed in the table contained within Figure 5.2, together with the variation in the perilune altitude during the 6-month propagation time. Of course, these initial conditions are not necessarily unique, but they supply insight into a combination of orbital elements that lead to each type of motion. Figure 5.3a-d also displays the paths traced out by the perilunes along these candidates in the $e - \omega$ polar plot. Each path is plotted with a unique color that matches the color in the first column of the table in Figure 5.2. For clarity, these 15 paths are separated across multiple plots based on their inclination.

Using the table in Figure 5.3 as a reference, each type of motion exists at a unique value of inclination in the Moon-fixed frame, consistent with previous analyses. For instance, Lara, Ferrer, and De Saedeleer identify low lunar frozen orbits with eccentricities between ~ 0 to 0.035 and initial arguments of perilune that place the initial perilune close to an axis of inertia, in particular



Inclination	Eccentricity	AoP	RAAN	Variation in altitude
🟧 i = 0.001°	0.020	240°	299.990°	11.9 km
🟩 i = 20°	0.005	90°	270.0°	60.9 km
🟦 i = 25°	0.005	120°	270.0°	50.2 km
🟥 i = 50°	0.005	30°	330.0°	13.6 km
🟪 i = 70°	0.020	240°	180.0°	23.4 km
🟨 i = 75°	0.005	0°	300.0°	31.1 km
🟪 i = 80°	0.035	90°	270.0°	18.4 km
🟦 i = 85°	0.005	270°	270.0°	24.7 km
🟦 i = 95°	0.010	270°	150.0°	17.9 km
🟥 i = 100°	0.020	90°	90.0°	27.2 km
🟪 i = 105°	0.015	210°	330.0°	15.4 km
🟥 i = 130°	0.005	150°	120.0°	22.6 km
🟪 i = 135°	0.040	270°	330.0°	29.8 km
🟦 i = 160°	0.005	180°	30.0°	90.4 km
🟨 i = 179.999°	0.020	210°	210.005°	34.9 km

Figure 5.2: Initial orbital elements used to generate the 15 candidates for low lunar frozen orbits at $a = 1838$ km at January 1, 2025 00:00.000 UTC in a 100x100 gravity field. Colors in the first column match the color scheme in Figure 5.3

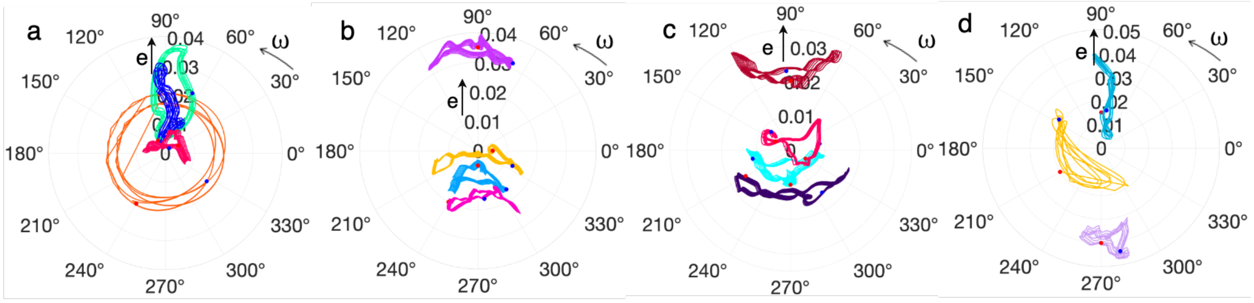


Figure 5.3: Evolution of perilune in the $e - \omega$ polar plot for the 15 candidates for low lunar frozen orbits at $a = 1838$ km at January 1, 2025 00:00.000 UTC in a 100x100 gravity field, with the red and blue dots indicating the initial and final states, grouped by initial inclination: a) from $i = 0.001°$ to $i = 50°$, b) from $i = 70°$ to $i = 85°$, c) from $i = 95°$ to $i = 130°$, and d) from $i = 135°$ to $i = 179.999°$

near the $\hat{\boldsymbol{Z}}$- or $\hat{\boldsymbol{Y}}$-axes, which are therefore associated with higher stability [61]. This result is also consistent with the conditions derived by Folta and Quinn using Lagrange's planetary equation, such that for $39.23° < i < 140.77°$ if $\omega = 90°$ or $\omega = 180°$ there is a value of eccentricity that drives $\dot{\omega}$ and $\dot{e}$ to zero [37]. Similar results have also been found using the averaged Hamiltonian equations with higher-order gravity field models [61].

Some of the candidates for low lunar frozen orbits in Figure 5.3a-d have been identified by previous authors, offering verification of the results presented in this work. For example, the candidates that exist at $75^\circ \leq i < 105^\circ$ match the frozen orbits presented in 2007 by Russell and Lara [96]. Lara also studied a frozen orbit at $i = 88^\circ$ with a similar perilune evolution to the candidate frozen orbit at $i = 85^\circ$ [60]. Furthermore, Park and Junkins use the Lagrange planetary equations in a low-fidelity gravity model to derive combinations of the average eccentricity and inclination of frozen orbits at $a = 1838$ km when $\omega = 90^\circ$ or $\omega = 270^\circ$ [89].

To further analyze the extracted candidates for lunar frozen orbits, the evolution of perilune is examined in the Moon-fixed frame. Specifically, Figure 5.4 displays a selection of the most bounded members of several clusters. Each subfigure displays a group of representatives that are clustered together in $\mathcal{C}_r$ during Step 5 of the presented framework. Trajectories with perilunes that predominantly exist in the northern hemisphere of the Moon are plotted in shades of blue, whereas those with perilunes mostly in the southern hemisphere are plotted in shades of red. In the right inset of each subfigure, the path traced out by the most bounded trajectory in the $e - \omega$ polar plot appears along with the initial inclination labeled for reference.

The evolution of perilune along the candidates for frozen orbits is compared in the Moon-fixed frame. Figure 5.5 displays the perilune location along each trajectory in the Moon-fixed frame using unique colors; these colors do not match previous figures. Figure 5.5a plots the perilune evolution for trajectories with a low inclination, as listed in the legend. Across this plot, the perilunes evolve with small variations in the latitude. Notably, trajectories with initial inclinations of $i = 1 \times 10^-3^\circ$ and $i = 179.999^\circ$ produce perilunes that are tightly bounded to a region that is slightly offset from the $-\hat{\boldsymbol{X}}$- and $+\hat{\boldsymbol{X}}$-axis, respectively. Figure 5.5b displays the perilune evolution of trajectories with a higher inclination. In this case, the perilunes evolve over a larger range of latitudes, with some completing two revolutions in the Moon-fixed frame every lunar rotational period.

Figures 5.4 and 5.5 reveal near antisymmetric properties in the candidates for frozen orbits as a function of inclination around $i = 90^\circ$. For instance, the trajectories with the most bounded evolution of perilune at $i = 20^\circ$ and $i = 160^\circ$ possess a similar geometry with perilunes occurring

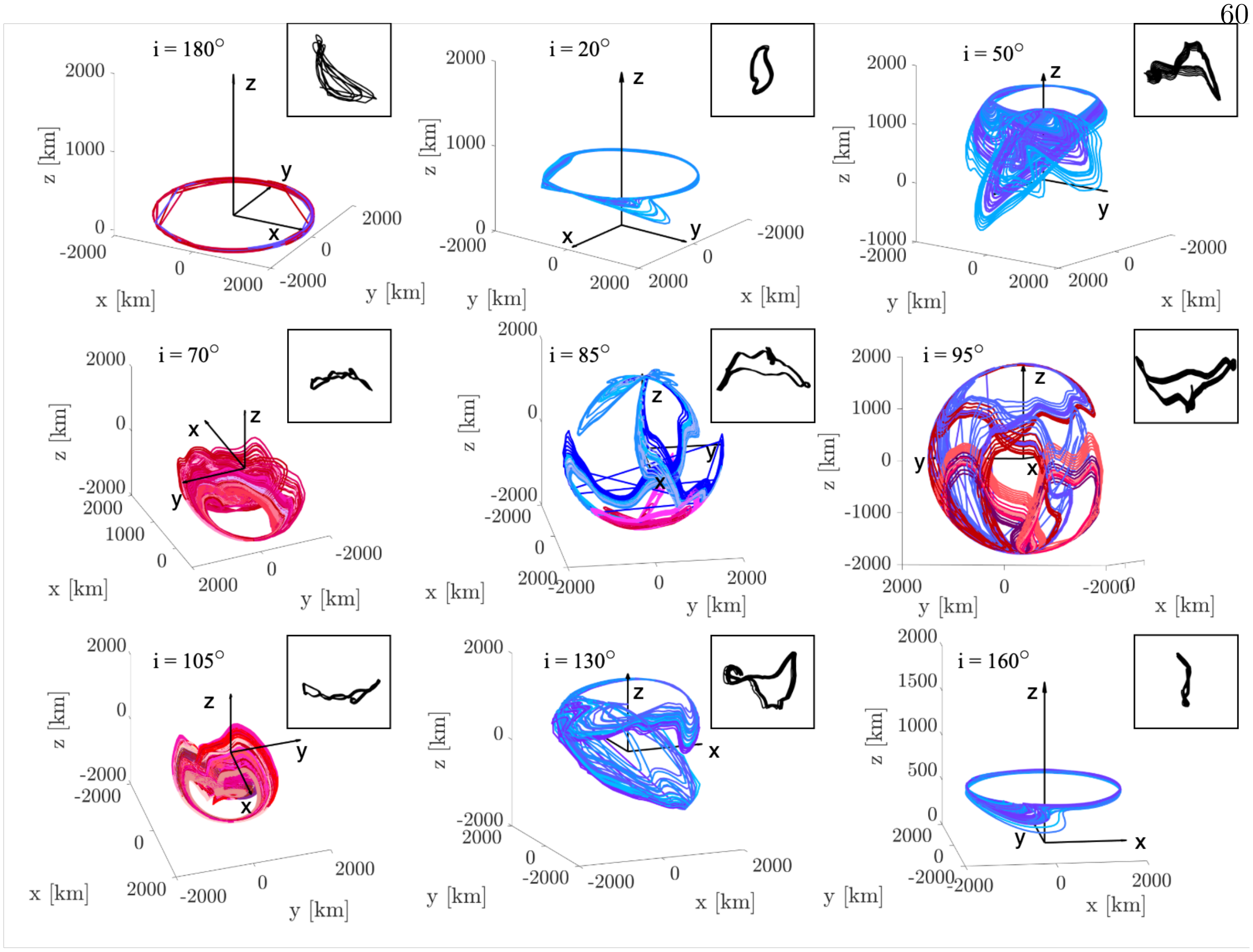


Figure 5.4: Perilune evolution in the Moon-fixed frame for selected representatives that are grouped together in $\mathcal{C}_r$ during Step 5 of the clustering-based framework

at near constant latitudes over the northern hemisphere of the Moon, except for one region where a ripple occurs. This ripple nearly antisymmetrically occurs over the $+\hat{\boldsymbol{X}}$ and $-\hat{\boldsymbol{X}}$ hemispheres of the Moon, respectively. A similar observation holds for the trajectories with $i = 20°$ and $i = 160°$. The trajectories at high inclinations exhibit a slightly different geometry of the perilune evolution in the Moon-fixed frame: each path performs two revolutions near the $+/-\hat{\boldsymbol{Z}}$-axis. However, for $i = 85°$, for example, this path spans the $+\hat{\boldsymbol{X}}$-hemisphere in the Moon-fixed frame, whereas for $i = 95°$, the perilunes exist predominantly over the $-\hat{\boldsymbol{X}}$-hemisphere. Aggregating these observations, the candidate lunar frozen orbits that are prograde and nonplanar produce perilunes that evolve most significantly over the $+\hat{\boldsymbol{X}}$ hemisphere, while orbits that are retrograde and nonplanar possess

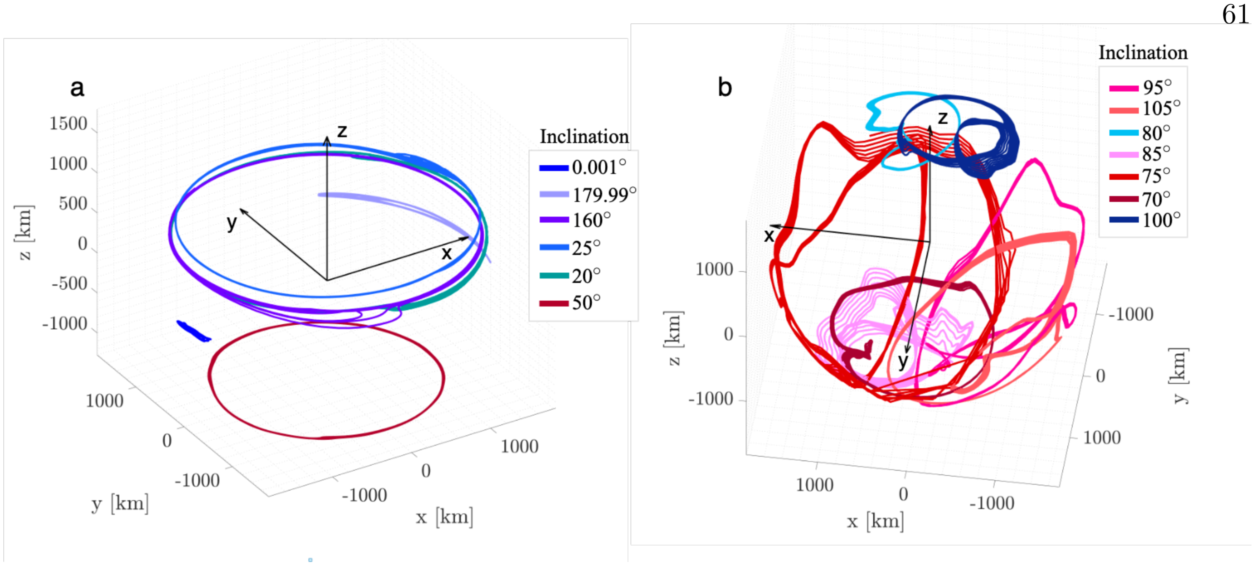


Figure 5.5: Evolution of perilune in the Moon-fixed frame for candidates for low lunar frozen orbits: a) low inclination trajectories and b) high inclination trajectories

perilunes that evolve most significantly in the $-\hat{\boldsymbol{X}}$ hemisphere. Although the $\hat{\boldsymbol{X}}$-axis is aligned with the mean direction to the Earth, a more extensive analysis of the dynamical contribution governing the characteristics of the perilune evolution in the Moon-fixed frame is the focus of ongoing work.

5.2 Evolution of Orbit Lifetime in each Cluster

Clusters of trajectories also support the identification of trends in the orbital elements that lead to changes in the orbit lifetime. Consider the 26 clusters that do not contain bounded motion and compare the evolution of trajectories across clusters at the same inclination when the eccentricity increases. The perilunes possess varying levels of drift and, as a result, orbit lifetimes. This drift tends to increase as the maximum eccentricity along the path increases. As an example, consider the time evolution of trajectories with an initial inclination of $i = 5°$ in Figure 5.6. For each trajectory, displayed on a single row, the following information is plotted from left to right: the evolution of perilune in the pq-plane; the variation in altitude over time, normalized by 180 days; the variation in inclination over time, normalized by 180 days; and the variation in eccentricity

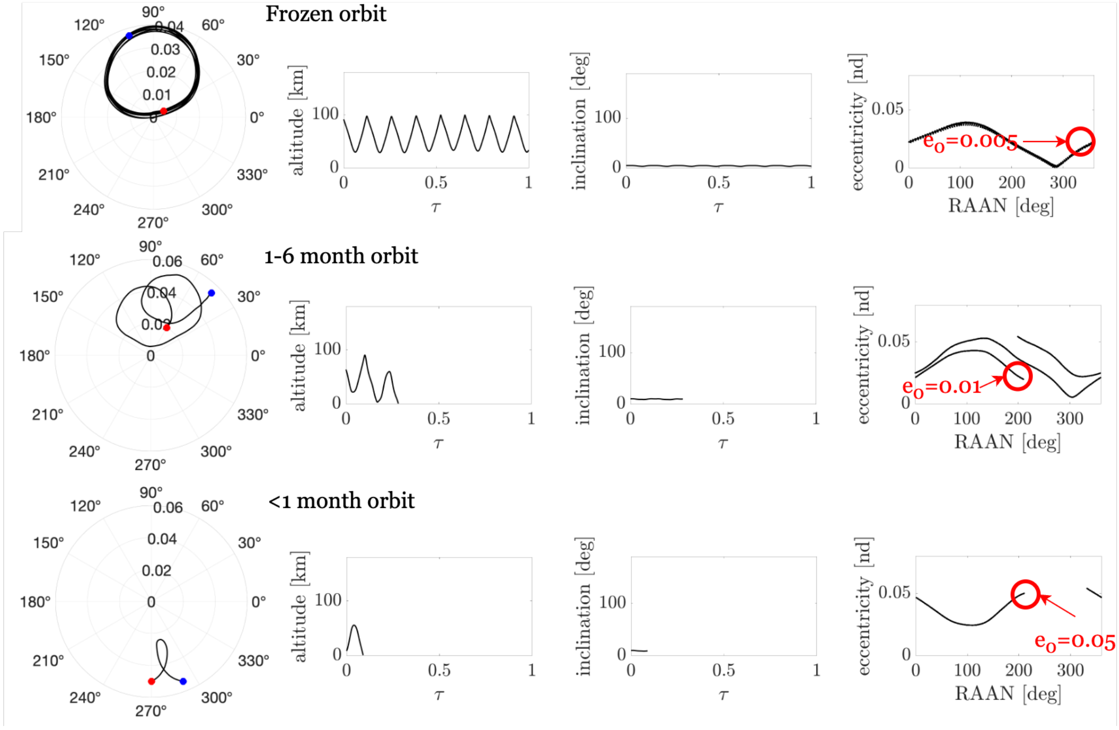


Figure 5.6: Comparing the evolution of the orbital elements for three orbits in a cluster at $i = 5°$: (top) frozen orbit, (center) orbit with 1-6 month lifetime, and (bottom) orbit with < 1 month lifespan

as a function of RAAN. At this inclination, the initial value of $e = 0.005$ produces a candidate for a frozen orbit, as displayed in the top row. However, as the initial eccentricity is increased to the values annotated in red in the right column of this figure, an increased drift occurs between the revolutions in the pq-plane over each lunar rotational period; simultaneously, the orbit lifetime decreases. At sufficiently high values of the eccentricity, the trajectory impacts the lunar surface in less than one lunar rotational period before completing a full revolution in the pq-plane.

Within $\mathcal{C}_g$, some clusters of trajectories with a lifetime of 180 days do not lead to frozen orbit candidates. Consider the representative trajectory of a cluster with $i = 90°$, plotted in Figure 5.7a in the $e - \omega$ polar plot along with the variation in its altitude, inclination, eccentricity, and RAAN. The perilune along this trajectory traces out a curve on average in the $e - \omega$ polar plot. This perilune evolution is similar to the evolution of paths that exist in the same cluster as a frozen orbit, e.g., in Figures 4.2 and 4.6. However, the perilunes along the trajectory in Figure 5.7a and its associated cluster members intersect the lunar surface before completing a full revolution. Furthermore, the center of these curves traced out by the drifting perilune paths does not exist at an eccentricity that lies below the critical value corresponding to lunar impact at $a = 1838$ km. However, frozen orbits with a geometrically similar perilune evolution to this trajectory have been observed by Folta and Quinn to exist at $i = 90°$ with a higher-semi-major axis of $a = 1861$ km [37].

The remaining 26 clusters within $\mathcal{C}_g$ that do not produce candidates for lunar frozen orbits include clusters of trajectories with a lifetime of less than 180 days. As an example, the evolution of perilune for members of two clusters is plotted in the pq-plane in Figure 5.7b with $i = 120°$ and 5.7c with $i = [160°, 180°]$. In Figure 5.7b, the perilunes of trajectories with a 1-6 month lifetime secularly drift towards the right in the pq-plane until impacting the lunar surface. Physically, this drift corresponds to the argument of perilune approaching 0° and, therefore, the perilune approaching a region around the mean direction to the Earth within the $\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}$ plane of the Moon-fixed frame. A similar secular drift in the perilune location within the pq-plane occurs in Figure 5.7c for a group of trajectories with an orbit lifetime of less than 1 month.

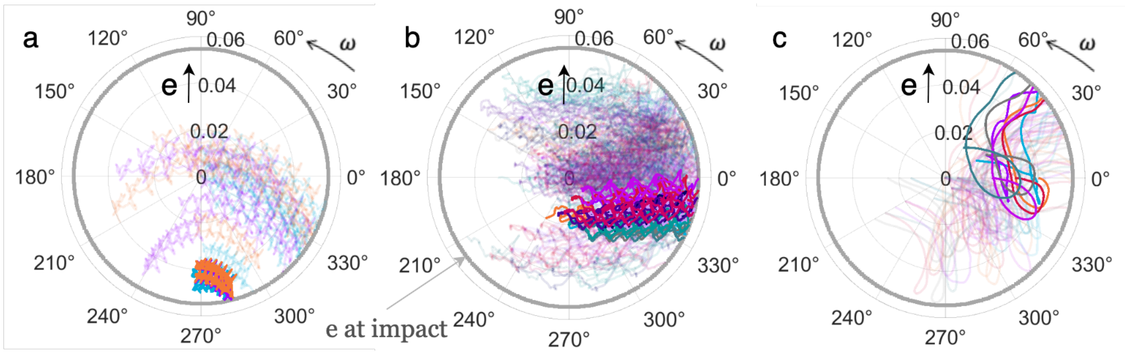


Figure 5.7: Evolution of perilune in the $e - \omega$ polar plot for two clusters of trajectories: a) with > 6 month lifetime but not bounded; b) with a 1-6 month orbit lifetime and c) with a <1-month orbit lifetime

Chapter 6

Motion Primitive Approach to Trajectory Design in Multi-Body System

This Chapter presents an updated motion primitive approach to trajectory design in a multi-body system. This version of the trajectory design framework is used throughout this chapter to generate trajectories between L_1 and L_2 Lyapunov orbits at different energy levels.

6.1 Motion Primitives in Robotics

In robotics, motion primitives are defined as the building blocks of motion [112] and have typically been extracted using manual labeling, clustering, or basis function approximations [39, 87, 109, 93]. Using motion primitives, a continuous-time path planning problem can be reformulated as a discrete graph search problem. For instance, Frazzoli, as well as Majumdar and Tedrake, have constructed graphs by defining the nodes as motion primitives and directed edges connecting selected nodes [39, 65]. Graph search algorithms may then be used to construct sequences of motion primitives that form a complex trajectory. This approach has been successfully applied to a variety of problems, ranging from describing and planning motion for humans or robots performing simple movement tasks [58, 27, 21] to path planning for autonomous air or land vehicles [65, 39, 58, 27, 21].

6.2 Motion Primitives in Spacecraft Trajectory Design

Inspired by the use of motion primitives in robotics, Smith and Bosanac introduced a motion primitive approach to spacecraft trajectory design in a multi-body system [101]. In their work, primitives are extracted as geometrically different arcs along fundamental solutions of the CR3BP.

Similar to the motion primitives in robotics, these arcs can be composed together to create a path connecting one state to another in the rotating frame, composing a trajectory initial guess. The first version of the motion primitive design approach consisted of six steps.

First, a segment of the solution space is summarized by constructing a library of motion primitives that have been extracted using consensus clustering. In their proof-of-concept, Smith and Bosanac used this clustering approach to summarize the geometric, stability, and energy characteristics of fundamental solutions such as periodic orbit families and their associated hyperbolic invariant manifolds in the Earth-Moon CR3BP [100]. Then, a motion primitive graph is constructed, with nodes defined as the motion primitives and directed edges reflecting the potential for sequential composability of the selected primitives. This graph was constructed in two steps. Initially, a set of subgraphs was used to summarize the sequential composability of primitives derived from the same family of fundamental solutions, i.e, the edges connecting nodes representing primitives of the same family are defined. Subsequently, the subgraphs are connected via a high-level itinerary graph, a structure that determines which and how the subgraphs are connected [101]. Subsequently, a depth-first search algorithm was employed to generate all possible primitive sequences of a specified length that connect a desired initial and target primitive [101].

Selected primitive sequences of specified lengths are then refined to improve the quality of the associated initial guess. In their work, Smith and Bosanac used a sequential trimming and morphing approach to trim the primitives and select the best member of the primitive set that would reduce the overall discontinuity of the initial guess [101]. Then, these initial guesses are corrected using collocation and multi-objective optimization to compute a transfer resembling the initial guess while reducing the maneuver requirements [101]. Finally, the continuous initial guesses serve as input to a continuation process, which produces a set of trajectories with gradually varying geometries, times of flight, and total Δv [101].

Although the motion primitive design approach was originally introduced by Smith and Bosanac [101], several substantial improvements are presented in this dissertation and detailed in the following sections.

6.3 Step 1: Generating Motion Primitives

Motion primitives are generated in the Neptune-Triton CR3BP to supply the building blocks of complex spacecraft trajectories. Similar to their use by Smith and Bosanac [100], motion primitives are generated in the NT-CR3BP to summarize continuous families of periodic orbits as well as arcs along the hyperbolic invariant manifolds of unstable primitives from selected periodic orbit families [72]. First, groups of orbits or arcs with a similar geometry are constructed to summarize each family of periodic orbits or set of hyperbolic invariant manifolds. Then, a motion primitive is defined as a single representative member of each group. This process is repeated across the selected set of fundamental solutions to produce a library of motion primitives that discretely summarize continuous trajectories across a segment of the phase space. Although this library of motion primitives provides sufficient information for the trajectory design problems addressed in this dissertation, ongoing work by other researchers involves expanding the library to encompass a more general set of trajectories that do not necessarily correspond to fundamental solutions.

6.3.1 Curvature

The concept of curvature from differential geometry is used in this work to create geometrically meaningful arcs along trajectories in the CR3BP [13, 41]. The value of curvature κ can be computed at any state along a trajectory as [91]

$$\kappa(t) = \frac{\sqrt{(\ddot{z}\dot{y} - \ddot{y}\dot{z})^2 + (\ddot{x}\dot{z} - \ddot{z}\dot{x})^2 + (\ddot{y}\dot{x} - \ddot{x}\dot{y})^2}}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}} \tag{6.1}$$

While the value history of curvature along a trajectory provides insights about its general shape, the maxima of this function identify the locations where the trajectory is most significantly curved. In particular, maxima in the curvature satisfy the following condition:

$$\begin{aligned}\dot{\kappa}(\boldsymbol{x}) = 0 = & \frac{2(\dddot{y}\dot{x} - \dddot{x}\dot{y})(\dot{x}\ddot{y} - \ddot{x}\dot{y}) + 2(\dddot{x}\dot{z} - \dddot{z}\dot{x})(\dot{z}\ddot{x} - \ddot{z}\dot{x}) + 2(\dddot{z}\dot{y} - \dddot{y}\dot{z})(\dot{y}\ddot{z} - \ddot{y}\dot{z})}{2(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2 + (\dot{y}\ddot{z} - \ddot{y}\dot{z})^2}} \\ & - \frac{3(2\dot{x}\ddot{x} + 2\dot{y}\ddot{y} + 2\dot{z}\ddot{z}))\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2 + (\dot{y}\ddot{z} - \ddot{y}\dot{z})^2}}{2(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{5/2}}\end{aligned} \tag{6.2}$$

where

$$\dddot{x} = 2\ddot{y} + \dot{x} - \frac{(1-\mu)\dot{x}r_1^3 - (1-\mu)(x+\mu)\dot{r}_1^3}{r_1^6} - \frac{\mu\dot{x}r_2^3 - \mu(x-1+\mu)\dot{r}_2^3}{r_2^6} \tag{6.3}$$

$$\dddot{y} = -2\ddot{x} + \dot{y} - \frac{(1-\mu)\dot{y}r_1^3 - (1-\mu)y\dot{r}_1^3}{r_1^6} - \frac{\mu\dot{y}r_2^3 - \mu y\dot{r}_2^3}{r_2^6} \tag{6.4}$$

$$\dddot{z} = -\frac{(1-\mu)\dot{z}r_1^3 - (1-\mu)z\dot{r}_1^3}{r_1^6} - \frac{\mu\dot{z}r_2^3 - \mu z\dot{r}_2^3}{r_2^6} \tag{6.5}$$

and $\ddot{\kappa} < 0$. These maxima often occur in similar locations to apses relative to meaningful reference points, with the advantage that they do not require the specification of a reference point [11]. Therefore, maxima in curvature are used to identify geometrically meaningful points along a trajectory spanning a wide region of the Neptune-Triton system.

6.3.2 Motion Primitives of Arcs Along Stable or Unstable Manifolds

The primitives from the hyperbolic invariant manifold are obtained from clusters of manifold trajectory arcs with similar geometric features. Bosanac developed the clustering procedure employed in this section, originally presented in Reference [12] and later adapted for motion primitives in Reference [41]. Clusters obtained from this approach group trajectory arcs with consistent shape and path locations, accurately summarizing the fundamental solutions of the system.

First, arcs are defined leveraging the concept of curvature [13]. After trajectories are generated along the selected stable or unstable manifold, the curvature κ is calculated as outlined in Section 6.3.1 at all the states along trajectories. Then, initial conditions for arcs are selected at each maximum in the curvature along these trajectories. The arcs are defined to include a specified number of maxima in the curvature. In this work, arcs span an $n_{\kappa_{max}} = 5$ maxima, and subsequent arcs have four overlapping maxima in curvature [41].

Successively, clustering is used to group arcs with similar geometry. Initially, each arc is described using two finite-dimensional feature vectors: the first captures the shape of the trajectory, and the second describes the positions it traverses in configuration space. Successively, each arc is discretized at the point of maxima in curvature, and then at two states that are equally spaced in arclength between pairs of subsequent maxima. Therefore, each trajectory arc is discretized at a total of $n_L = 13$ states. Given the samples, the shape-based feature vector $\boldsymbol{\nu}_v$ is defined as

$$\boldsymbol{\nu}_v = [\hat{\dot{x}}_1, \hat{\dot{y}}_1, \hat{\dot{z}}_1, ..., \hat{\dot{x}}_{n_L}, \hat{\dot{y}}_{n_L}, \hat{\dot{z}}_{n_L}]^T \tag{6.6}$$

where $[\hat{\dot{x}}_i, \hat{\dot{y}}_i, \hat{\dot{z}}_i]$ is the velocity unit vector at the ith sample in $\mathcal{R}$. While the position-based feature vector is defined as

$$\boldsymbol{\nu}_p = [x_1, y_1, z_1, ..., x_{n_L}, y_{n_L}, z_{n_L}]^T \tag{6.7}$$

where $[x_i, y_i, z_i]$ is the position vector at the ith sample in $\mathcal{R}$. These feature vectors produce a $3n_L$-dimensional description of the arc.

The feature vectors describing the arcs along each manifold trajectory are clustered to identify groups of geometrically similar trajectories. This clustering step is performed in two stages, utilizing two clustering algorithms. First, the Hierarchical Density-Based Spatial Clustering of Applications with Noise [17] is leveraged to retrieve coarse clusters of trajectories with similar shapes. Then, the coarse clusters are refined using the Density-Based Spatial Clustering of Applications with Noise algorithm, considering both shape and position feature vectors.

In the first step, N_t trajectories are sampled in n_L nodes to create the shape-based feature vectors. This creates a $N_t \times 3n_L$ matrix of features, which is then input to HDBSCAN. Additionally, the algorithm requires the specifications of hyperparameters n_{size}, n_{sample}, and ϵ_{merge}, as discussed in Section 3.2. At this step, $n_{size} = 5$ and $n_{sample} = 4$, whereas $\epsilon_{merge} = 2\sqrt{13}\ \sin(5°)$ [13]. Together, small values of n_{size} and n_{sample} prioritize grouping trajectories with localized variations across the dataset. On the other hand, ϵ_{merge} is computed as the Euclidean distance between a sequence of 13 unit vectors that are separated by an angle of 10°. This process produces $\mathcal{C}_C$ coarse groups that contain trajectory arcs with similar geometrical evolution.

In the second step, the coarse clusters are refined following the approach originally developed by Bosanac [12], which uses a convoy detection scheme to ensure that each cluster comprises only members with consistent shapes that exist in a similar configuration space location. In this refinement approach, each coarse cluster is evaluated independently. First, the values of the shape-based feature vector $\boldsymbol{\nu}_v$ and position-based feature vector $\boldsymbol{\nu}_p$ are extracted at each ith sample.

Considering a cluster composed of N_{cg} members, this produces an $N_{cg} \times 3$-dimensional matrix for each type of feature vector, $\boldsymbol{D}_p$ and $\boldsymbol{D}_v$.

Secondly, the matrices $\boldsymbol{D}_p$ and $\boldsymbol{D}_v$ at all the ith samples are input independently to DBSCAN to produce $2n_L$ clustering results. From the $2n_L$ clustering results, the refined clusters are generated as follows: if any trajectories are consistently clustered together in all $2n_L$ clustering results, then they are considered similar; however, if at one point, the trajectories are assigned to different groups, they are considered dissimilar and cannot be assigned to the same cluster. Trajectories existing in clusters with at least n_{size} members will form a refined cluster; otherwise, they will be labeled as noise and discarded. This process can potentially maintain a single unique cluster, remove outliers and noise points, or split a cluster into multiple smaller clusters. This process is repeated for all $\mathcal{C}_C$ coarse groups to produce a set of $\mathcal{C}_R$ refined clusters.

Additional inputs to DBSCAN are the hyperparameters n_{sample} and ϵ, as discussed in Section 3.1. In this dissertation, these values are selected as $n_{sample} = 4$ and $\epsilon = (n_{sample} + 1) \max(e, \epsilon_{threshold})$, where $\epsilon_{threshold}$ is a minimum threshold and e is the n_{size}-largest distance between each member and its nearest neighbor. This heuristic is modified compared to Reference [12] based on several factors. First, scaling the neighbors' distance, rather than the actual n_{sample}-neighborhood radii, minimizes the presence of outliers in the cluster. Secondly, the n_{size}-largest nearest neighbor distance helps reduce the impact of outliers while avoiding excessive splitting of clusters or misclassification of noise during refinement. Finally, by introducing the $(n_{sample} + 1)$ scaling factor, the estimate of the n_{sample}-neighborhood size incorporates some margin.

Consistent with the original concept by Smith and Bosanac [101], motion primitives are extracted as a single representative member of each cluster. The medoid of the cluster in the position-based feature vector space of the entire trajectory is selected as the representative member and can be computed as

$$\boldsymbol{\nu}_r = \arg \min_{i \in \{1,..,N_t\}} \left(\sum_{\substack{j=1 \\ j \neq i}}^{N_t} ||\boldsymbol{\nu}_i - \boldsymbol{\nu}_j|| \right) \tag{6.8}$$

Then, $N_{\mathbb{R}_e} = 50$ trajectories in the clusters are sampled at an equal distance from the primitive to obtain a representation of the region of existence ($\mathbb{R}_e$) of that motion type in the solution space.

An example of the geometrically different clusters obtained with this approach is shown in Figure 6.1, both for stable and unstable invariant manifolds of an L_1 Lyapunov orbit. The primitives are highlighted with thick lines, while the light-colored region surrounding them represents the region of existence. In each subfigure, the gray or blue circles indicate Neptune and Triton, scaled by their equatorial radii of 24,764 km and 1,352.6 km, respectively. Then, the circle along the trajectory indicates the arc's initial state, and red diamonds locate nearby equilibrium points. This same figure configuration is used throughout this work when visualizing selected motion primitives. The primitives and their regions of existence are stored in a motion primitive library, which offers both a condensed, discrete representation of the continuous solution space and supports the construction of complex trajectories. These clusters, as well as other clusters used in this dissertation, were generated by Dr. Natasha Bosanac and are used in the trajectory design problems presented in this and the following chapter.

6.3.3 Motion Primitives of Periodic Orbit Families

Different from the hyperbolic invariant manifolds, motion primitives from periodic orbits are obtained by considering the full trajectories and not splitting them into arcs. Previous versions of this work have shown the applicability of clustering techniques such as WEAC [100, 101, 74] and HDBSCAN [73, 72] to identify primitives from periodic orbit families. However, this process can sometimes lead to the creation of motion primitives with little geometrical difference or a strong dependence on the density of the samples along a continuous family. Therefore, this work employs an analytical approach to detect changes in the continuous evolution of a periodic orbit family, thereby identifying the fundamental, geometrically distinct members of the set.

Changes in geometry within an orbit family can be computed analytically without resorting to unsupervised learning approaches. This approach, previously presented by Gillespie, Miceli, and Bosanac [41], leverages the evolution of curvature along a nonlinear trajectory to capture its shape within the rotating plane. First, all the members of periodic or resonant orbit families are

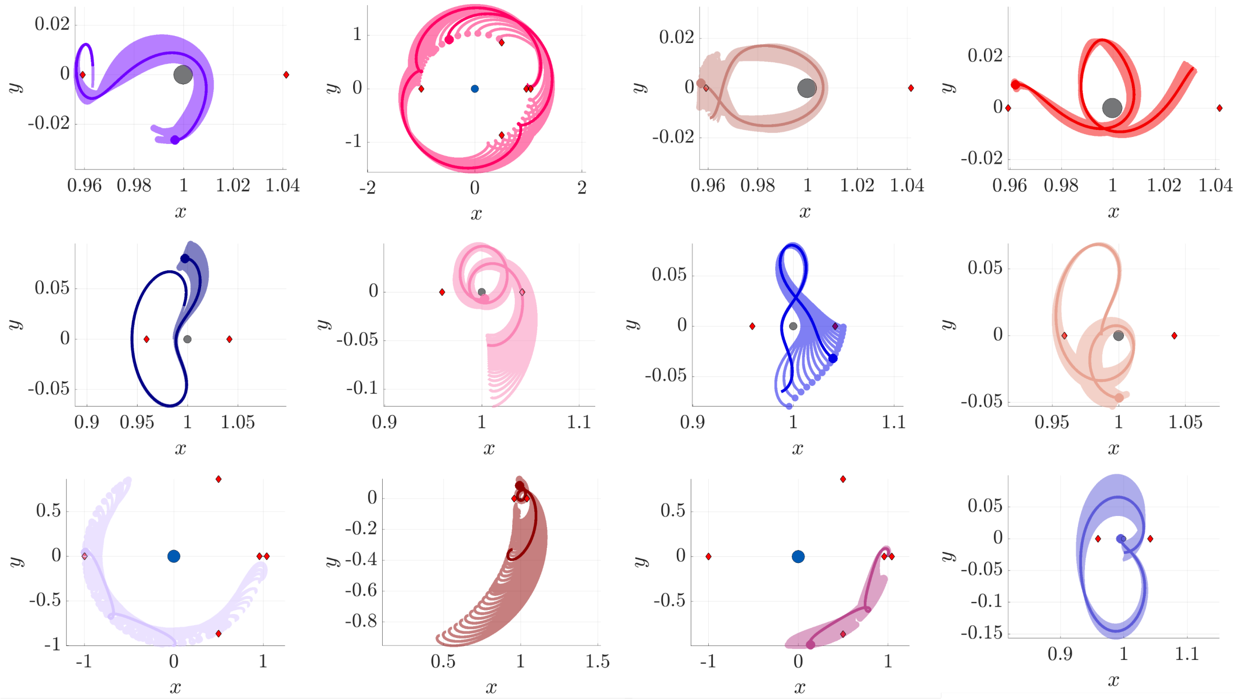


Figure 6.1: Selected motion primitives (thick curves) and their region of existence (shadowed area) for the L_1 Lyapunov orbit family at $C_J = [2.9981, 3.0037, 3.01400]$. The dot represents the initial state of each trajectory

propagated to compute the maxima in curvature as described in Equation 6.1. Then, the change of maxima in curvature points is monitored along the family. Subsequent orbits with the same number of curvature maxima are considered similar and assigned to the same group; then, when the number of curvature points changes, a new group is formed. The change in curvature points is an indicator of a geometric change within the family; therefore, it is leveraged to automatically group trajectories that exist in a continuous set, such as a periodic orbit family. Once the groups of similar orbits are found, primitives are extracted as the medoid of each set, together with $N_{\mathbb{R}_e} = 20$ trajectories that represent the primitive's $\mathbb{R}_e$.

To demonstrate this process, consider motion primitives extracted to summarize selected planar periodic orbit families. Each subfigure in Figure 6.2 displays motion primitives and the associated regions of existence for the following families: a) L_1 Lyapunov orbits, b) DPOs, and c) a

3:1 resonant orbit family. In each figure, each color corresponds to a group of geometrically similar periodic orbits: the thick curve traces out the periodic orbit selected as the motion primitive, whereas the associated region of existence is depicted using the light-colored region in the same shade. Analysis of this figure reveals that this approach yields sets of motion primitives that capture both obvious and subtle geometric changes along each family of periodic orbits.

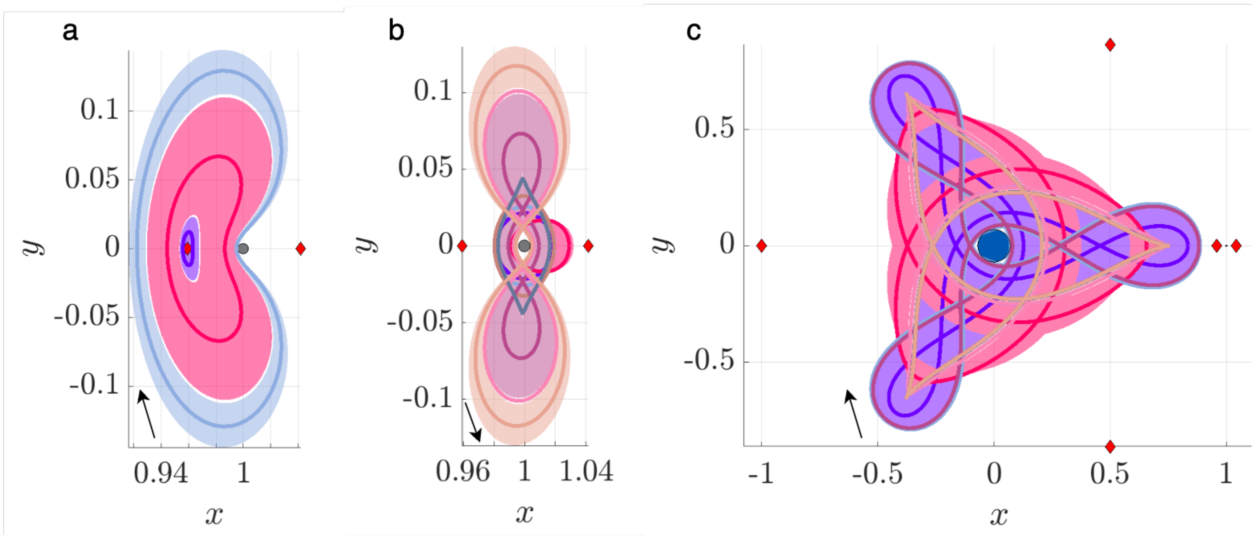


Figure 6.2: Selected motion primitives (thick curves) and their regions of existence (translucent shading) for the a) L_1 Lyapunov orbit family, b) distant prograde orbits, and c) a 3:1 resonant orbit family

6.3.4 Summary of Procedure

Step 1 focuses on generating motion primitives that summarize the diverse array of geometries exhibited across families of periodic orbits and arcs along the hyperbolic invariant manifolds of selected periodic orbits. A graphical, high-level overview of the technical approach described in Sections 6.3.2 and 6.3.3 is presented in the flowchart in Figure 6.3. Several parameters and decisions govern the motion primitives generated in this step, summarized in Table 6.1.

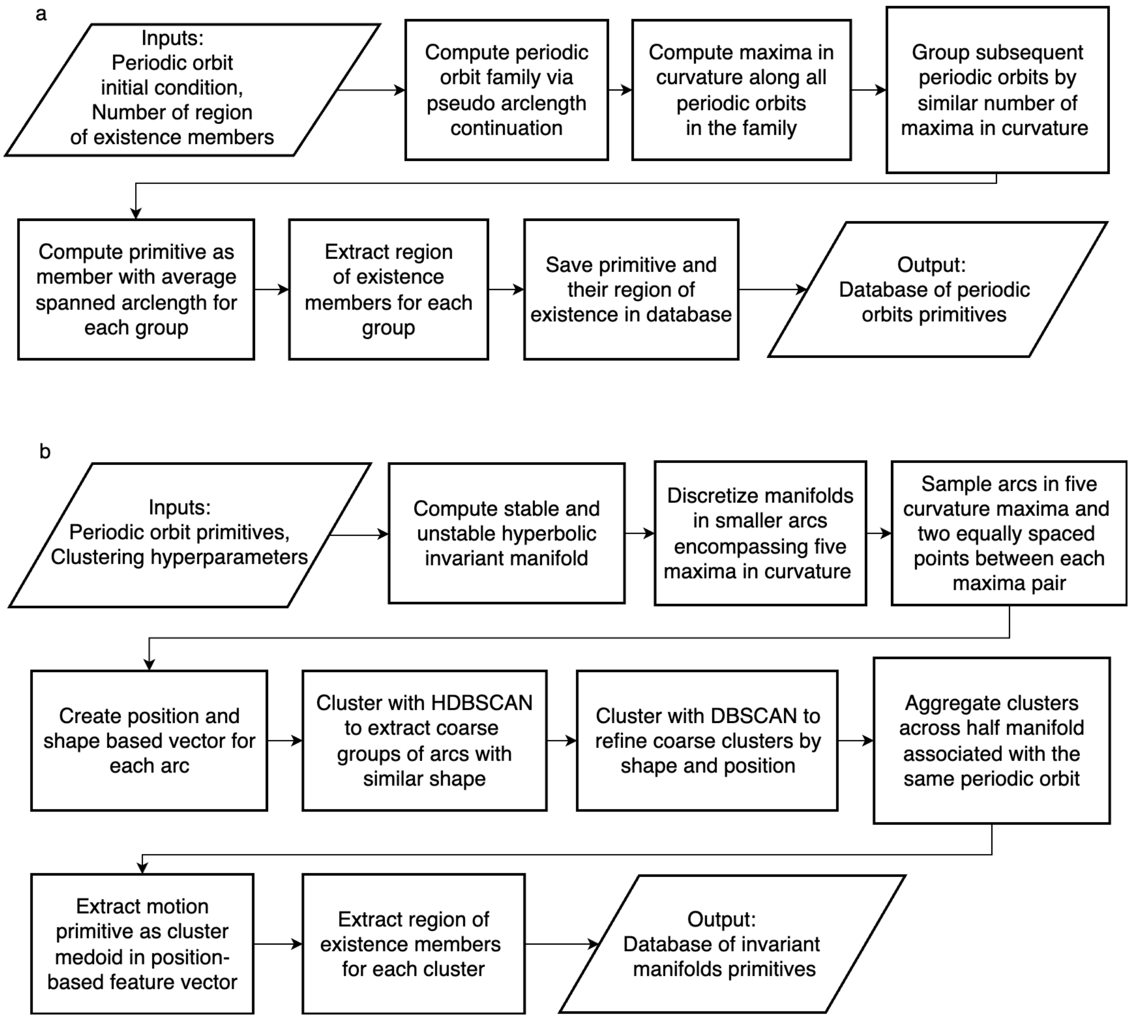

a
Inputs:
Periodic orbit
initial condition,
Number of region
of existence members
Compute periodic orbit family via pseudo arclength continuation
Compute maxima in curvature along all periodic orbits in the family
Group subsequent periodic orbits by similar number of maxima in curvature
Compute primitive as member with average spanned arclength for each group
Extract region of existence members for each group
Save primitive and their region of existence in database
Output:
Database of periodic orbits primitives
b
Inputs:
Periodic orbit primitives,
Clustering hyperparameters
Compute stable and unstable hyperbolic invariant manifold
Discretize manifolds in smaller arcs encompassing five maxima in curvature
Sample arcs in five curvature maxima and two equally spaced points between each maxima pair
Create position and shape based vector for each arc
Cluster with HDBSCAN to extract coarse groups of arcs with similar shape
Cluster with DBSCAN to refine coarse clusters by shape and position
Aggregate clusters across half manifold associated with the same periodic orbit
Extract motion primitive as cluster medoid in position-based feature vector
Extract region of existence members for each cluster
Output:
Database of invariant manifolds primitives

Figure 6.3: Flowcharts of process used to generate a) motion primitives from periodic orbits and b) motion primitives from hyperbolic stable and unstable invariant manifolds.

6.4 Step 2: Constructing a Motion Primitive-Informed Graph

The trajectory design process begins by relating the motion primitives through the definition of a 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 [85]. 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 they represent. In previous applications of path-planning with motion primitives, both in robotics and astrodynamics, graphs

Governing parameter	Influence	Value
Arc definition	Defines region described by the motion primitive and influences the geometric representation of the solution space	Encompassing 5 maxima in curvature
Sampling scheme	Characterizes the low-dimensional representation of the trajectory arcs.	13 samples: 5 at maxima in curvature and 2 equally spaced in arclength between consecutive maxima in curvature
Feature vector	Influences how trajectory arcs are compared to assess similarities. Can be used to highlight a specific property more than another, e.g., position, shape, time of flight, etc.	position $\boldsymbol{\nu}_p$ and velocity direction $\boldsymbol{\nu}_v$
Roe size	Describes the area where similar geometries exist and influences the initial guess refinement by providing more options to move across primitives	50 members from the manifold cluster and 20 members for periodic orbit groups, equally spaced in position from the primitive
Clustering algorithm	Influence how arcs with similar features are grouped and distinguished from dissimilar arcs	HDBSCAN for coarse clustering and DBSCAN for refinement
Clustering parameters	Govern the clustering algorithms and influence their results, and as a consequence, the primitives	$n_{size} = 5$, $n_{sample} = 4$, $\epsilon_{merge} = 2\sqrt{13}\ \sin(5°)$ and $\epsilon = (n_{sample} + 1)\max(e, \epsilon_{threshold})$

Table 6.1: Table of governing parameters for motion primitive generation

have been formulated to assign distinct motion primitives as nodes and directed, weighted edges to describe the connectivity between the primitives [39, 65, 101, 74, 72].

This step of the technical approach builds upon the previous approach presented by Smith and Bosanac [101] to include several improvements. First, primitives are not represented as a node of the graph, but instead, they inform the definition of the graph components. The graph's nodes are now defined as states along the primitives, while the graph's edges are added only between nodes closely located in phase space and are weighted by the velocity direction discontinuity between the two nodes. Additionally, constraints can be imposed during the graph construction to incorporate path or maneuver constraints. This new definition of the motion primitive graph results in a better description of the primitives' sequential composability and thus, better recovery of initial guesses that could support the design of constrained trajectories.

6.4.1 Selecting Motion Primitives for the Graph

Primitives potentially suitable for the design are selected from the database generated in Step 1. Among all available motion primitives, selection can be guided by several factors. In this work, a primitive is considered suitable if its geometry and direction of motion resemble those of the boundary primitives of the transfer and if it falls within the Jacobi constant range defined by the initial and final targets. When no prior knowledge is available regarding the type of transfer or its boundary conditions, any or all primitives in the database may be included in the design. However, this may result in graphs that yield no valid solution and require significant computational effort to generate. When a specific geometry or energy range is identified, the database is searched for the periodic orbit families or hyperbolic invariant manifolds that have members that meet the desired characteristics. Then, motion primitives are selected from the periodic orbit families or the hyperbolic invariant manifolds to which they belong. For example, up to three motion primitives could be selected from the L_1 Lyapunov family shown in Figure 6.2a, depending on the energy and geometry that is required from the design.

The selected primitives and the members of their $\mathbb{R}_e$ might span a large area of the state space. However, mission or scientific constraints might require the spacecraft to fly at a specific distance from a body or to avoid certain regions of the system. To address this concern, path constraints can be added to modify the extension of a primitive region. Specifically, these constraints are formulated as the minimum or maximum distance from a primary and can be applied to all or some of the selected primitives within the graph [72]. At this stage, as the graph represents a discrete approximation of the solution space, the values of the constraint should be considered with a margin. For example, if the mission requires a minimum allowable altitude from the surface of Triton of $h_{min} = 500\ km$, this constraint can be imposed on the graph step as $h_{min} = 300\ km$. Then, the initial guesses resulting from a constrained graph can be corrected using the actual constraint value of 500 km.

The primitives selected for the design are analyzed to check if they meet the conditions imposed by the constraint. 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 is removed from the graph. Otherwise, if only some trajectories are flagged, the region of existence is simply resized to contain only the members that meet the constraints. If the primitive of the set is flagged, then another primitive is identified among the remaining trajectories. The primitives and the members of its $\mathbb{R}_e$ that satisfy the constraint are input for the graph generation.

6.4.2 Identifying Sequentially Composable Motion Primitives

In this work, unlike previous applications of motion primitives in path planning [39, 65, 101, 74, 72], motion primitives do not constitute the nodes of the graph. Although such an approach could lead to the recovery of geometrically diverse trajectories as demonstrated by Smith and Bosanac [101], these graphs do not sufficiently capture the sequential composability of the primitives. When reducing a set of states, i.e., a primitive, to a single node, the definition of an edge cannot be generalized to any couple of nodes, but must take into account the properties of each primitive. In their earlier work, Smith and Bosanac proposed four different types of edges to accommodate that need [101]. The user could select the most appropriate edge between two nodes based on the knowledge of the primitives, e.g., primitives derived from the same manifolds would be connected considering the last sample along the first primitive and any sample along the second primitive. Overall, the construction of the graph required knowledge of the properties of primitives and the manual definition of the types of edges to use between specific primitives. Therefore, extending the approach to larger trajectory design problems and connecting diverse types of primitives extending across the configuration space could be challenging and time-consuming.

To eliminate the need for manual graph setting, the definition of the edge has been generalized to encompass any pair of nodes. In this case, an edge can be constructed considering any state along the primitives represented by the nodes. However, the paths recovered using this definition of nodes and edges would often lead to unfeasible primitive sequences. To demonstrate this concept, consider the definition of the previous motion primitive graph over a generic set of arcs, without specifying the type of edge to use between pairs of primitives. First, motion primitive arcs are discretized in a set of samples to account for their geometrical evolution. Each primitive is considered a node of the graph. Then, edges are allowed between any pair of primitives, and the minimum state discontinuity between a pair of samples along two primitives is taken as the weight of the edge connecting two nodes. However, such an edge definition does not reflect the actual composability of a sequence of primitives. Figure 6.4 provides an example of a connectivity issue that arises with this approach. In Figure 6.4a, P_1 (in maroon) and P_2(in lilac) are connected considering the state at sample 5 along P_1 and the state at sample 6 along P_2. In the same figure, P_2 is connected to P_3 (in blue) using the state at sample 2. The sequence of primitives $P_1 - P_2 - P_3$ in 6.4b obtained after trimming the primitives in sequential order results in an infeasible path because the connection between P_2 and P_3 is defined at a state that exists temporally "before" the state at which P_1 and P_2 are connected.

To address the sequentiability problem, this work formulates a motion primitive graph to capture the feasible sequences of motion primitives in the NT-CR3BP. In this approach, the samples along each motion primitive become nodes in the graph, and the edges represent connections between composable nodes. For instance, each primitive sampled at M_s states evenly distributed along the arc length contributes to M_s nodes in the graph. Accordingly, a graph capturing N_{mp} motion primitives includes $N_{mp} \cdot M_s$ nodes. Therefore, the motion primitives are not nodes of the graph, but rather inform the construction of the graph.

Edges between nodes are added only if the composability requirements are met. First, the regions of existence of the two primitives at the corresponding samples must overlap or be within

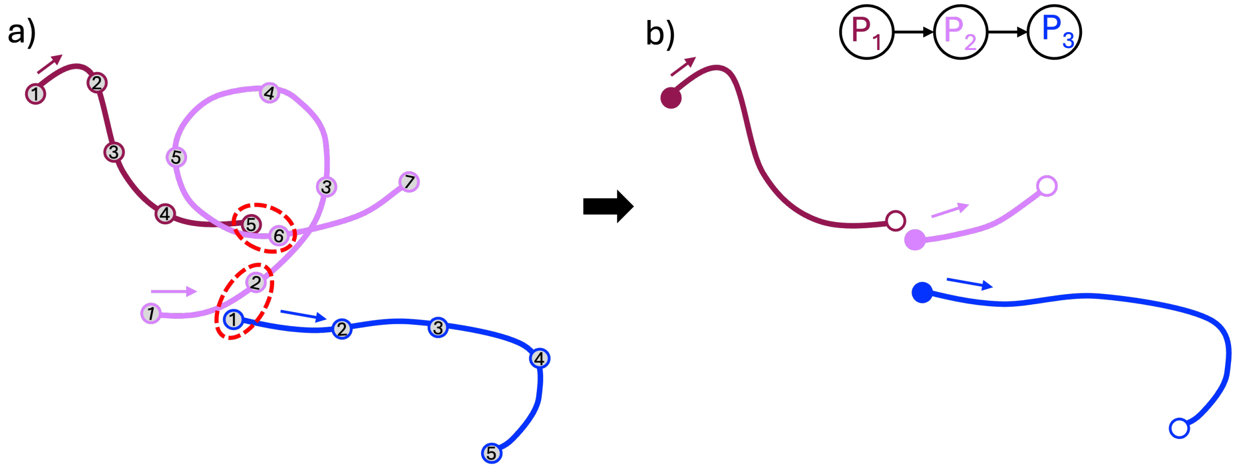


Figure 6.4: Example of a primitive sequence when considering primitives as a node of a graph. a) P_1 (in maroon) and P_2 (in lilac) are connected at samples 5 along P_1 and 6 along P_2; while P_2 and $P3$ (in blue) are connected at samples 2 along P_2 and 1 along P_3. b) the primitive sequence obtained by linking the primitives at the samples used to create the graph's edges. This graph cannot consider the order of the samples used to create the edges, resulting in a discontinuous sequence.

close proximity. This condition is evaluated in two steps. First, the radius of position proximity around each sample, labeled $e_{a,b}$, where a is the primitive number and b is the sample number, is computed considering the average inter-sample distance between all the members in the $\mathbb{R}_e$ at that sample. Then, the pairwise distance between each sample along the two primitives $\mathbb{R}_e$ is computed and compared to the proximity region of the samples. If the distance between two samples is less than the sum of the position proximity radius $e_{a,b}$, then the samples are considered to be in position proximity. For example, considering the fourth sample along the first primitive P_1 and its region of existence $\mathbb{R}_e(P_1)$ in light blue in Figure 6.5, then the radius of close proximity at that sample is computed as

$$e_{1,4} = c \cdot \frac{\sum_{i=1}^{U_{N_1}-1} \boldsymbol{r}_{1,i+1,4} - \boldsymbol{r}_{1,i,4}}{N_{\mathbb{R}_e(P_1)}} \tag{6.9}$$

where $N_{\mathbb{R}_e}$ is the number of trajectories in $\mathbb{R}_e(P_1)$ and c is a user-defined constant set to $c = 1.5$, while U_{N_1} is the total number of elements in $\mathbb{R}_e(P_1)$. Then, considering the first sample along P_2, the same value can be computed to obtain $e_{2,1}$ as shown in lilac in Figure 6.5. The distance between the fourth sample in $\mathbb{R}_e(P_1)$ to the first sample in $\mathbb{R}_e(P_2)$ is evaluated to obtain the matrix $d(\boldsymbol{r}_{1,k,4} - \boldsymbol{r}_{2,l,1})$, with dimension $N_{\mathbb{R}_e(P_1)} \times N_{\mathbb{R}_e(P_2)}$. If any of these pairwise distances satisfy the condition

$$d(\boldsymbol{r}_{1,k,4} - \boldsymbol{r}_{2,l,1}) \leq e_{1,4} + e_{2,1} \tag{6.10}$$

then the fourth sample of P_1 is considered in close proximty with the first sample of P_2. In Figure 6.5, the intersecting spheres indicate that the fourth sample along P_1 and the first sample in P_2 meet the proximity constraints.

Successively, nodes in position proximity are analyzed to check composability in velocity. In general, nodes are allowed to have a discontinuity in velocity magnitude to recover initial guesses, allowing for the placement of maneuvers. However, initial guesses characterized by abrupt changes in direction of motion and large energy difference between the arcs composing the paths would lead to unfeasible or expensive trajectories. Therefore, the nodes' composability conditions include limitations on the velocity discontinuity between two nodes. Among the pair of nodes satisfying the position proximity condition, the discontinuity in velocity magnitude, Δv, and velocity direction, $\Delta \hat{\boldsymbol{v}}$, is assessed. The pair of nodes must satisfy

$$\begin{gathered} \Delta \hat{\boldsymbol{v}} \leq \theta_{max} \\ \Delta v_{min} \leq \Delta v \leq \Delta v_{max} \end{gathered} \tag{6.11}$$

where θ_{max} is the change in velocity direction between the two nodes which is set to $\theta_{max} = 30°$, while the values Δv_{max} and Δv_{min} can be specified if desired. If constraints on velocity magnitude are not included at this step, this condition is automatically satisfied for all nodes that meet the velocity direction change constraint.

Pairs of nodes that satisfy all conditions are considered composable and are connected with directed, weighted edges. The outputs of the composability check include the indices of the composable nodes along two primitives, as well as the members of the regions of existence U_i of the two primitives where those nodes are composable. The nodes are used to generate the edges in the primitive informed graph, while the indices of the $\mathbb{R}_e$'s trajectories are used in the search algorithm described in Step 3.

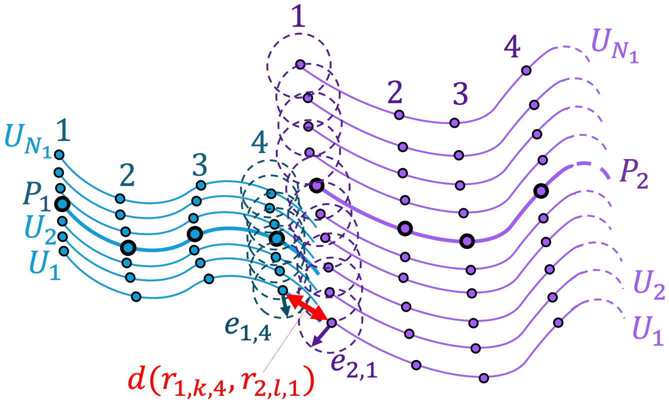


Figure 6.5: Schematic representation of composability assessment between two primitives and the trajectories in their region of existence

6.4.3 Defining the High-Level Graph Structure

In this work, the motion primitive graph is initialized as a high-level itinerary graph, as presented by Smith and Bosanac [101]. After selecting the appropriate primitives for the design, these are assigned to sets that define the order in which they can be leveraged during the path search. Sets are typically defined in the following order: the Start, the Transfer, and the Target sets. The Start set contains the primitives representing the trajectory's initial conditions, while the Target set contains its destination. The Transfer sets group all primitives that can be leveraged to reach the target primitives from the initial ones. Multiple Transfer sets can be used to incorporate an itinerary via waypoints. With this definition, high-level itineraries can be defined with three or more sets as required.

The connectivity between and within the sets defines the direction of the edges between composable nodes along the primitives. In this work, the primitives of one set are connected to all the primitives of the following set; that is, all the members of the Start set are connected to all the members of the Transfer set. Then, the primitives within the same set are all interconnected; that is, primitives belonging to a family of the Transfer set can be connected to primitives of another family within the set, and vice versa. Moreover, primitives within the same family are also allowed to be interconnected. These connectivity choices allow maximum flexibility in the primitive exploration between and within the sets. However, different connectivity strategies can be selected depending on the trajectory design case. If two primitives are connected, then the nodes in the graph corresponding to those primitives can be connected by an edge, if the composability conditions illustrated in section 6.4.2 are met.

An example of a high-level itinerary graph is provided in Figure 6.6. Here, the primitives in one block are connected with the primitives in the next block on the right. The black symbol at the top right of the Transfer set indicates that the different primitives within the set are interconnected.

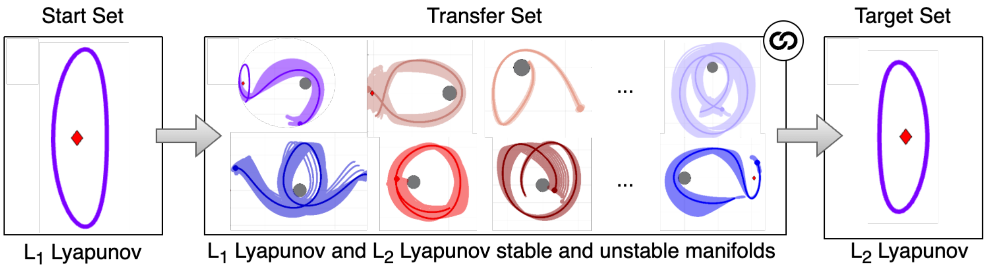


Figure 6.6: Example of high-level itinerary graph for designing a transfer between an L_1 Lyapunov orbit at $C_J = 3.01400$ and L_2 Lyapunov orbit at $C_J = 3.01377$, using the primitives from their unstable and stable manifolds

6.4.4 Constructing the Nodes of the Graph

The nodes of the graph are defined as samples along the primitives. In this work, all the primitives are sampled based on the curvature approach used in the primitive extraction process described in Step 1. Thus, each primitive obtained from the hyperbolic invariant manifolds generates $n_L = M_{s,m} = 13$ nodes, where five states are located at maxima in curvature and two samples are located at equal arclength intervals between each pair of maxima. For the primitives obtained from periodic orbits, the total number of nodes added to the graph from each primitive $M_{s,p,i}$ is variable and depends on the total number of curvature maxima along the orbit. The arcs between maxima encompassed by periodic orbits, such as the exterior resonant orbits, are characterized by a high arclength, which can often span more than a quarter of the position space. Therefore, periodic orbits are sampled by considering five equally spaced samples in arclength between pairs of maximum points. The increased number of samples, in this case, produces a more accurate representation of the primitive. Overall, if the motion primitive-informed graph includes $N_{mp,m}$ motion primitives that are generated from arcs along stable or unstable manifolds and $N_{mp,p}$ primitives from periodic orbits, then the total number of nodes composing the graph is $N_{mp,m}M_{s,m} + \sum_{i=1}^{N_{mp,p}} M_{s,p,i}$. Each node is labeled based on its position in the high-level itinerary and the corresponding family member, allowing it to be easily recognized when analyzing the initial guesses in Steps 3 and 4.

6.4.5 Constructing the Edges of the Graph

Directed edges are added only between composable nodes, as discussed in Section 6.4.2, and between nodes along the same primitive to represent how the motion can flow across and along the primitives. To represent the smoothness of transitioning from one primitive to another, the edges are weighted by a positive scalar that encodes the change in velocity direction between two nodes. Between the reachable states $\boldsymbol{r}_{a,k,i}$ and $\boldsymbol{r}_{b,l,j}$ along two primitives P_a and P_b, the edge weight is defined as

$$q_{i,j} = \max\left[(1 - \hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{v}}_j), 10^{-15}\right] \tag{6.12}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the velocities at the nodes $\boldsymbol{r}_{a,k,i}$ and $\boldsymbol{r}_{b,l,j}$.

Note, however, that the nodes belonging to the same primitives are connected with quasi-zero weight $q_{i,i+1} = 10^{-15}$ to describe that traveling along the same primitive doesn't require a maneuver. This edge weight is not set to zero to avoid confusion with the description of missing edges, which are represented by the value of 0 in the adjacency matrix representing the graph.

An example of this edge construction step between two primitives is depicted conceptually in Figure 6.7. Composable samples are shown in red in Figure 6.7a. The nodes in the graph corresponding to the red samples are connected with weighted edges. Otherwise, if no samples are composable in Figure 6.7b, then no edge is created between the nodes corresponding to these

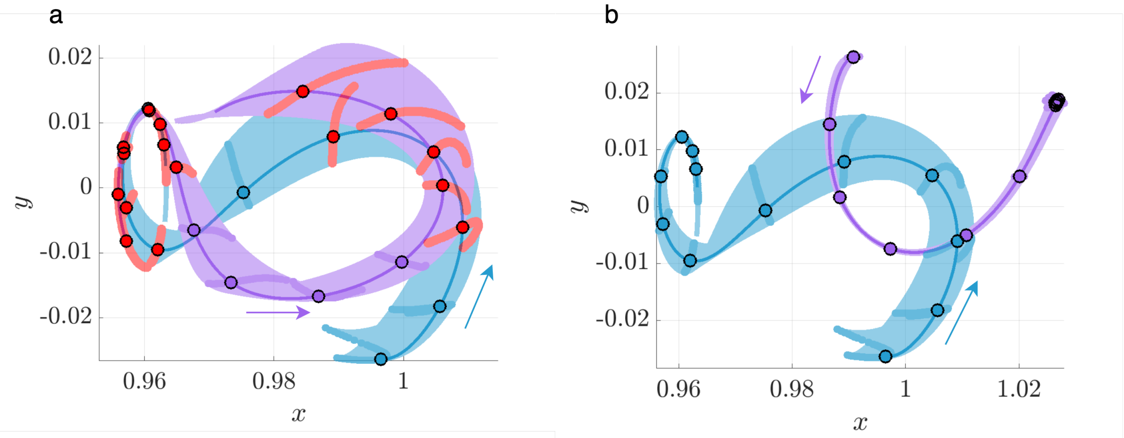


Figure 6.7: Example of: a) Primitive's region of existence intersecting along the red nodes leading to edges placement between correspondent nodes in the graph; and b) non-intersecting primitive regions that do not produce edges between nodes in the graph

samples in the graph. Compared to the prior version of this approach, this motion primitive graph better captures potential connectivity between two primitives. This approach adaptively adds edges as needed, eliminating the need for manually selecting the number or types of edges. The graph generated from the high-level itinerary in Figure 6.6 using the approach described in this section consists of 200 motion primitives, which correspond to 1003 nodes and 12058 edges.

6.4.6 Summary of Procedure

Step 2 focuses on generating a motion primitives graph that summarizes the sequential composability of the primitives selected for the design. A graphical, high-level overview of this approach is presented in the flowchart in Figure 6.8. Several parameters and decisions governing this step are summarized in Table 6.2.

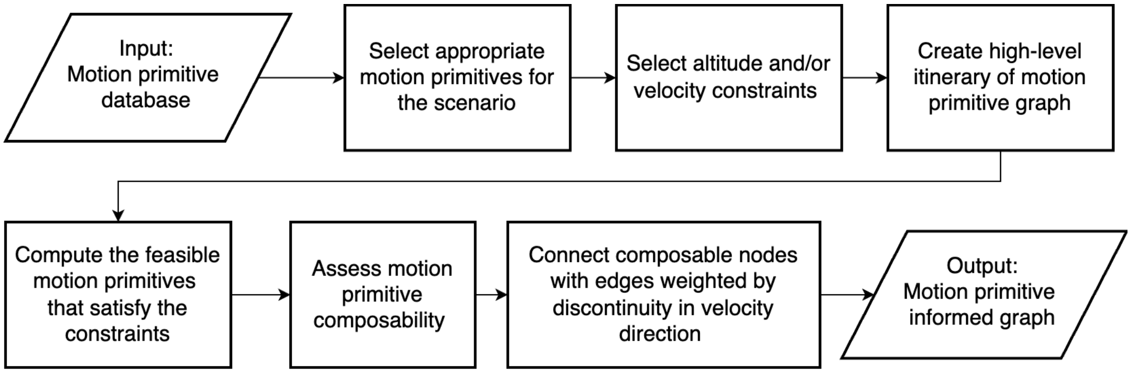

Input:
Motion primitive database
Select appropriate motion primitives for the scenario
Select altitude and/or velocity constraints
Create high-level itinerary of motion primitive graph
Compute the feasible motion primitives that satisfy the constraints
Assess motion primitive composability
Connect composable nodes with edges weighted by discontinuity in velocity direction
Output:
Motion primitive informed graph

Figure 6.8: Flowcharts of the process used to generate motion primitive informed graph

6.5 Step 3: Generating Sequences of Motion Primitives

Consistent with path-planning literature, motion primitives can be composed in sequences to form complex paths [112, 44]. Accordingly, Smith and Bosanac [101] searched the motion primitive graph to produce motion primitive sequences to support the initial guess construction process. However, rapidly discovering diverse primitive sequences with low total edge weight cost and diverse sequence lengths is challenging. Accordingly, this work presents a custom algorithm for searching the motion primitive graph. This algorithm is inspired by two well-known path planning algorithms, A* and Yen's algorithm [48, 113], but includes modifications to the original algorithms to ensure that the resulting paths exhibit sufficient diversity. Overall, this new approach can find a larger variety of solutions, producing smoother and diverse initial guesses in a more computationally efficient time.

6.5.1 Overview of Graph Search Algorithms

Considering a graph composed of nodes and directed, weighted edges, graph search algorithms can be used to solve the shortest-path problem, i.e., the problem of finding the sequences of nodes that minimize the sum of the edge weights, given the desired initial and final nodes. Many algorithms, including Dijkstra's algorithm [28] and A* [48], have been developed to find the shortest-path problem and have been successfully used in a variety of path-planning applications.

Governing parameter	**Influence**	**Value**
Nodes	Govern the discrete representation of the primitives in the state space	Sample states maxima in curvature and at 2 or 5 nodes equally spaced in arclength between consecutive curvature maxima
Edges	Represent the potential composability between primitives	Connecting only composable nodes and nodes along the same primitive, i.e., nodes that exist in position proximity and that satisfy the Equation 6.11, as described in Section 6.4.2
Edge weights	Define the value of state discontinuity between two composable nodes	Defined as in Equation 6.12
Composability parameters	Influence the identification of composable nodes based on position and velocity discontinuity	For position composability the constant value multiplied to the average node distance is $c = 1.5$, for velocity discontinuity in Equation 6.11 $\theta_{max} = 30°$ and $\Delta v_{max/min}$ may be included depending on the scenario
High-level itinerary connections	Influences how primitives are connected between different sets and within the same set	All the primitives in one set are connected to all the primitives of the next set, and all the primitives within one set are connected

Table 6.2: Table of governing parameters for constructing the motion primitive-informed graph

However, graphs might provide a variety of useful but sub-optimal solutions to a single-source single-target path planning problem. To address the necessity of exploring additional diverse paths in a graph, a problem commonly known as the k-shortest paths problem [86], other algorithms have been developed. These include Yen's algorithm [113], Eppstein's algorithm [31], and the Bellman–Ford algorithm [6, 38, 75].

Many k-shortest paths search algorithms leverage Dijkstra's algorithm to find the single best solution in a graph. Dijkstra's algorithm searches the shortest path between two nodes in a weighted graph. Starting from the initial node, the algorithm computes the cost $g_{i,j}$ to reach each jth neighboring node from the current ith node and selects the node associated with the minimum cost as the next node to explore. This process is repeated for the subsequent nodes until the destination node is reached or all the reachable nodes have been visited. Therefore, the result is the path in the graph that minimizes the sum of the edge weights [28].

While Dijkstra's algorithm is accurate as it explores paths based on their actual cost, it can be computationally expensive, especially for large graphs. Therefore, A* algorithm is preferred in many applications. A* was first introduced by Hart, Nilsson, and Raphael [48] in 1968 to prove that a heuristic function could be incorporated into the formal mathematical theory of graph searching and achieve optimality when compared to other search algorithms. The difference from Dijkstra's algorithm is in fact that at each iteration, A* explores the neighboring nodes by computing a cost $f_{i,j} = g_{i,j} + h_j$, which is the sum of the cost to go from the current node to the next, $g_{i,j}$, and the heuristic cost to go from the next node to the target node, h_j. All the incomplete paths up to the current node are stored in a queue with their associated temporary costs f_t and g_t. When the exploration of the closest nodes is complete, the algorithm orders the list of current paths so that, at the successive iteration, the path that is expanded is the one that minimizes the total expected cost of the path. Once the target node is reached, the search is concluded. The selection of the heuristic depends on the application of the graph. However, it can be proved that given a heuristic such that $h_j \leq h_j^*$, A* always returns the least expensive path from the start to the goal node, where h_j^* is the true or optimal cost to go from the current node to the goal node. By including this heuristic, the A* search algorithm produces a shortest path through the graph with lower computational complexity and in less time than the well-known Dijkstra's algorithm [86].

To solve the k-shortest paths problem, the search algorithm must be able to identify the overall best path in the graph, as well as $k - 1$ diverse paths with a sub-optimal cost. Yen's algorithm is used to identify k loopless paths within a directed weighted graph. At each iteration, the best solution within the graph is identified using Dijkstra's algorithm and stored in a permanent list $\mathcal{A}$. From the latest best path composed on N_n nodes, $N_n - 1$ subgraphs are created by removing one edge from the initial path at a time from the original graph. On each subgraph, Dijkstra's algorithm is used to identify the best path starting from the root, i.e., the sequence of nodes composing the best path up to the removed edge, to the target node. The paths obtained from each subgraph are saved in a temporary list $\mathcal{B}$. This list is ordered to rank the paths in terms of cumulative edge weight cost. The path with the lowest cost is then saved as the new best path and moved to list $\mathcal{A}$. List $\mathcal{B}$ is never emptied, thus the saved paths are the global optimal paths in the graph. This process is repeated until the k-best or all the possible paths existing in the graph are identified [113].

6.5.2 Searching the Motion Primitive Graph

The motion primitive search algorithm presented in this section is used to rapidly identify k-best diverse sequences of nodes. The nodes correspond to states along the primitives; thus, the paths identified on the graph can be translated into sequences of primitives. The search algorithm used in this step is based on A* and Yen's algorithms; however, 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 yields paths that are highly similar to one another and often closely resemble the best overall path. These solutions have limited interest in many applications [20] and do not provide a sufficiently representative view of the solution space summarized by the graph. Consequently, in this work, Yen's algorithm and A* are modified to ensure that a sufficiently diverse set of paths is discovered using a wider variety of motion primitives. The algorithm used to search the motion primitive-informed graph requires two steps: 1) generate one primitive sequence from the graph, and 2) generate an additional $k - 1$ primitive sequence.

The first step is obtained with this primitive-based A* described in Algorithm 1. In this approach, the cost $g_{i,j}$ is the value of the graph edge weight $q_{i,j}$, while the heuristic h_j is selected to equal the true cost to reach the target node from any node j, ensuring that the relationship

$h_j \leq h_j^*$ is always respected. This heuristic is computed using MATLAB's distances algorithm [68] to generate the shortest path from node j to the target with Dijkstra's algorithm. The A* algorithm is preferred over Dijkstra's algorithm because the use of a perfect heuristic and path pruning makes it faster, thereby speeding up the computation of k paths through Yen's algorithm.

Algorithm 1 Modified A* for Searching an Motion Primitive-Informed Graph

```
1: Initialize a queue at the start node. Each queue entry contains:
     • a sequence of nodes (an incomplete path),
     • $n_c$ the current node (last node in the path),
     • values of $f_t$ and $g_t$,
     • indices of composable trajectories in the region of existence of a primitive at the current
       node.
2: while current node $\neq$ target node do
3:     Sort the queue in increasing order of $f_t$.
4:     Select the incomplete path with the lowest $f_t$. The current node is the last node of the
       selected path.
5:     Identify all neighboring nodes connected to $n_c$.
6:     Filter out neighbors not connected through the same trajectory within the region of existence
       of a primitive.
7:     Remove neighbors already present in the current path to avoid cycles.
8:     for each remaining neighboring node do
9:         Run Algorithm 2.
10:    end for
11:    Rank neighbors by increasing $f_t$.
12:    Select neighbors to explore.
13:    for each selected neighbor do
14:        Create a new path by appending the neighbor to the current path.
15:        Add the new path to the queue with updated $g_t$, $f_r$, and composable trajectory indices.
16:    end for
17: end while
```

In a traditional implementation of A*, step 9 of Algorithm 1 would always involve selecting all the available neighbors. However, to encourage the generation of diverse paths while still identifying node sequences with relatively low-velocity discontinuities, this work modifies the logic used to select the neighboring node as described in Algorithm 2. For each case described in Algorithm 2, the algorithm ensures the following conditions:

- Line 3: Encourage exploration of the current primitive and avoid reduction of a primitive arc to a single node.

- Line 5: Supports increasing the diversity of paths soon after the initial condition, although the associated node may be suboptimal with respect to f_t.
- Line 8: Prioritizes the natural flow along a primitive while making sure the least expensive path is explored.
- Line 10: Ensures that additional n_p incomplete paths are added to the queue for future consideration, improving exploration of the solution space.

Algorithm 2 Neighbor Selection Strategy

```
1:  **Input:** Current node $n_c$, list of neighboring nodes $N(n_c)$, primitive information, initial condition state, cost function $f_t$, parameter $n_p$
2:  **if** $n_c$ is the first visited node along the current primitive **then**
3:      Select the next node along the primitive as the only neighbor to explore
4:  **else if** $n_c$ is part of the nodes along the starting primitive or arc **then**
5:      Select one neighbor at random from $N(n_c)$
6:      Explore only the selected neighbor
7:  **else if** lowest-cost neighbor corresponds to the next state along the current primitive **then**
8:      Select and explore only the lowest-cost neighbor
9:  **else**
10:     Randomly select $n_p$ neighbors from $N(n_c)$, ensuring inclusion of the lowest-cost neighbor
11:     Explore the selected $n_p$ neighbors
12: **end if**
13: **for** each $j$th neighbor **do**
14:     Compute $g_{i,j}$, $h_j$ and $f_{i,j}$
15: **end for**
```

To identify an additional $k-1$ diverse paths through the directed weighted graph, Yen's algorithm is slightly adapted to the characteristics of the motion primitive-informed graph. The modification of Yen's algorithm in line 7 removes all edges between nodes along the same primitive, rather than a single edge between subsequent nodes, when constructing each subgraph. Accordingly, if the latest best path is composed of N_n nodes that correspond to a sequence of N_{mp} motion primitives, $N_{mp}-2$ subgraphs would be created at each iteration of Yen's algorithm since the first and last primitive are never removed. Moreover, paths searched on the subgraphs are not constrained to start from the root of the last best path, as in the original Yen's algorithm [113].

Another difference from the traditional implementation of Yen's algorithm is the emptying of the list $\mathcal{B}$, containing the paths obtained from the subgraphs after each iteration. This choice promotes the selection of a more diverse array of paths, albeit at the slight compromise of path optimality. This search algorithm produces k diverse, yet relatively low-cost paths through the graph that connect the initial and target nodes. A pseudo-code of the approach is provided in Algorithm 3.

Algorithm 3 k-Best Paths Search Algorithm

```
Run Algorithm 1 to calculate the globally optimal path in the motion primitive-informed graph
Save the primitive sequence and cumulative edge weight in permanent list $\mathcal{A}$
while number of paths in $\mathcal{A} < k$ do
    Let $\pi$ be the most recently added path in $\mathcal{A}$
    Extract the list of $N_{mp}$ primitives from $\pi$
    for each primitive $P_i$ in the list of $N_{mp}$ primitives do
        Remove all edges in the graph connecting nodes in $P_i$ to create subgraph $G_i$
    end for
    In parallel, run Algorithm 1 on each subgraph $G_i$ to calculate a path from the start to goal nodes
    Save each new primitive sequence and cumulative edge weight into temporary list $\mathcal{B}$
    Select the path in $\mathcal{B}$ with the lowest cumulative edge weight
    Add this path to $\mathcal{A}$
    Empty list $\mathcal{B}$
end while
```

Once a path through the graph has been identified, the node sequence is transformed into a primitive sequence to support efficiently generating a diverse set of initial guesses. As an example, consider a sequence of 10 nodes: the first four nodes are associated with states along motion P_1, the next three nodes are associated with states along motion P_2, and the final three nodes are associated with states along motion P_3. This node sequence, generated from the graph search, is then translated into the primitive sequence $P_1 - P_2 - P_3$. While the primitive sequence does not explicitly define the connecting arcs between P_1, P_2, and P_3, it implies the presence of a set of states exhibiting low discontinuity across the respective regions of existence $\mathbb{R}_e(P_1)$, $\mathbb{R}_e(P_2)$, and $\mathbb{R}_e(P_3)$. This information is used in the refinement process to generate the most optimal path across the regions of existence, considering all the trajectories in these regions.

For the trajectory design used as an example in this Chapter, the motion primitive-based search algorithm is applied to the graph to generate 50 primitive sequences connecting the starting

L_1 Lyapunov to the target L_2 Lyapunov. The computational time required for this step strictly depends on the size of the graph, but can be the order of hundreds of seconds. Then, these sequences of primitives are used to obtain 50 trajectory initial guesses, which are then corrected and optimized to obtain continuous trajectories as presented in Steps 4 and 5.

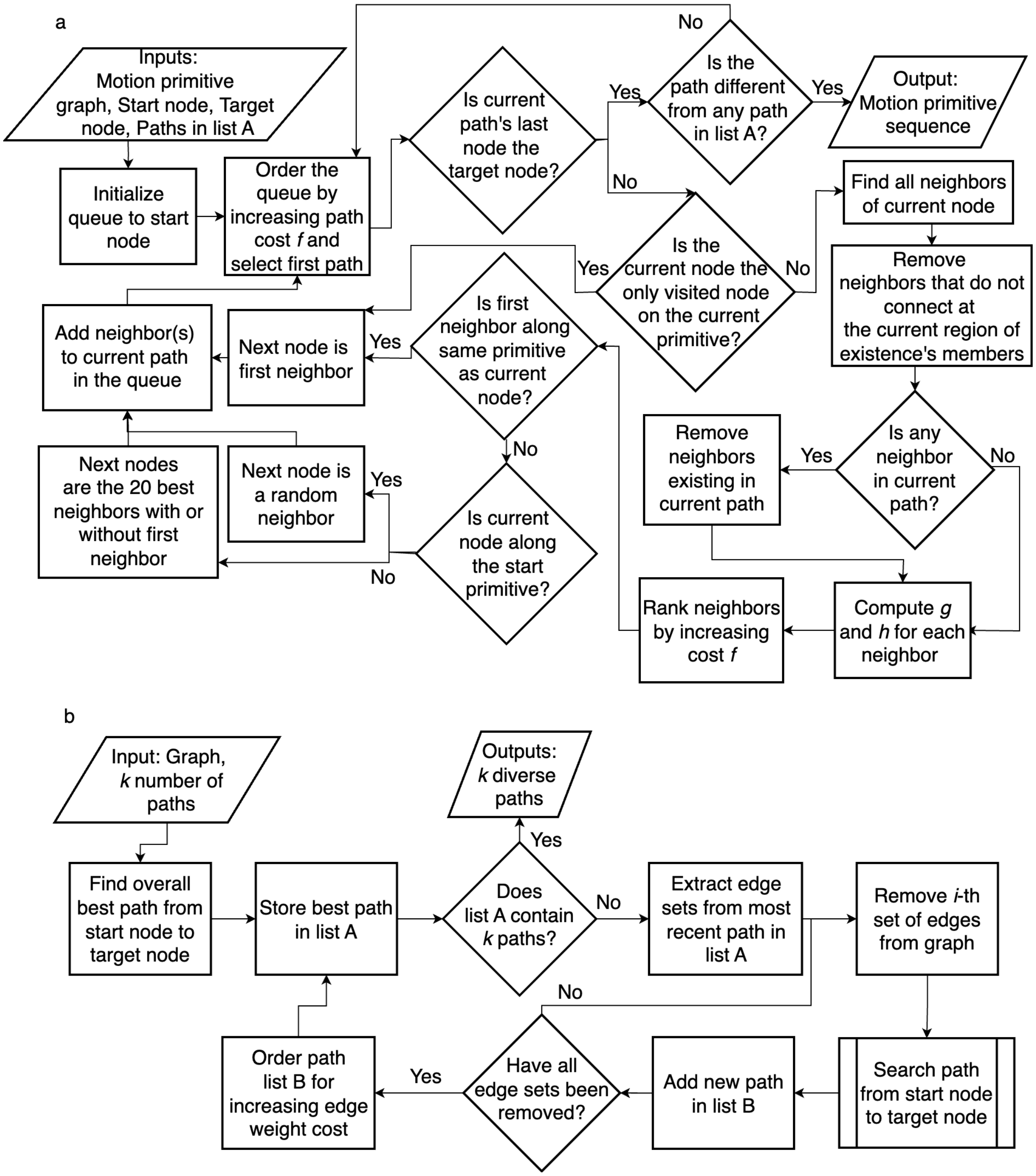

a
Inputs: Motion primitive graph, Start node, Target node, Paths in list A
Initialize queue to start node
Order the queue by increasing path cost *f* and select first path
Is current path's last node the target node?
Yes
No
Is the path different from any path in list A?
No
Yes
Output: Motion primitive sequence
Is the current node the only visited node on the current primitive?
Yes
No
Find all neighbors of current node
Remove neighbors that do not connect at the current region of existence's members
Is any neighbor in current path?
Yes
No
Remove neighbors existing in current path
Compute *g* and *h* for each neighbor
Rank neighbors by increasing cost *f*
Is first neighbor along same primitive as current node?
Yes
No
Next node is first neighbor
Add neighbor(s) to current path in the queue
Is current node along the start primitive?
Yes
No
Next node is a random neighbor
Next nodes are the 20 best neighbors with or without first neighbor
b
Input: Graph, *k* number of paths
Find overall best path from start node to target node
Store best path in list A
Does list A contain *k* paths?
Yes
No
Outputs: *k* diverse paths
Extract edge sets from most recent path in list A
Remove *i*-th set of edges from graph
Search path from start node to target node
Add new path in list B
Have all edge sets been removed?
No
Yes
Order path list B for increasing edge weight cost

Figure 6.9: Flowcharts of search algorithms used to generate a) one sequence of motion primitives and b) a diverse set of k optimal motion primitive sequences

6.5.3 Summary of Procedure

Step 3 describes the search algorithm developed to return diverse primitive sequences that minimize the edge weights over the motion primitive graph. A graphical, high-level overview of this approach is presented in the flowchart in Figure 6.9. Several parameters and decisions governing this step are summarized in Table 6.3.

Governing parameter	**Influence**	**Value**
Start node	Defines the beginning of the motion primitive sequence	First sample along the primitive representing the initial conditions of the transfer.
Goal node	Defines the end of the motion primitive sequence	Last sample along the primitive representing the target conditions of the transfer.
Number of best paths	Defines the number of diverse motion primitives initial guesses to extract from the graph.	Depends on the design scenario and a variable computational time.
Graph's edge weights	Influences which primitive is selected for the path	Minimum discontinuity in velocity directions at composable nodes.
Search algorithm	Identifies the best sequence of nodes that minimizes the sum of the edge weights	Custom algorithm that rapidly identifies diverse motion primitive sequences.
A* heuristic	Influences the way A* prioritizes the next node to explore.	h_i is equal to the exact cost to go from the next node to the target node.
Number of neighbors to explore in A* n_p	Influences how many neighboring nodes are explored at each iteration before arriving at the target node.	$n_p = 1$ if the next node is along the same primitive as the current node or if the current node is along the primitive representing the transfer's initial conditions, otherwise $n_p = 20$.
Starting and ending trajectory index	Indicates the member of the primitive's region of existence from which the path starts and ends. It is included in the queue when considering the reachable trajectory in step 6 in Algorithm 1	Index of the primitive trajectory.

Table 6.3: Table of governing parameters for motion primitive based search algorithm

6.6 Step 4: Constructing an Initial Guess for a Trajectory

The k-best initial guesses, corresponding to sequences of motion primitives, are refined to obtain arcs that minimize the state discontinuity and improve the likelihood of successfully recovering a nearby solution via differential corrections. This process is formulated using a second graph that includes only the sequence of primitives composing each path, with the members of their $\mathbb{R}_e$. This refinement graph is then searched to generate the sequence of arcs that minimize the velocity discontinuity. This approach provides a globally optimal, refined path, enhancing the original refinement process presented by Smith and Bosanac, which is based on sequential morphing and trimming [101]. This process is composed of three steps as depicted conceptually in Figure 6.10.

6.6.1 Defining the High-Level Graph Structure

First, each initial guess is considered as a sequence of motion primitive sets, where each set is composed of the motion primitive and the members of its $\mathbb{R}_e$, as displayed in Figure 6.10a. 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 one motion primitive set is connected to every member of the subsequent set to allow every possible combination within the primitives and the trajectories in $\mathbb{R}_e$. This step is displayed in Figure 6.10b.

Considering, for example, a primitive sequence generated in Step 3, $P_1 - P_2 - ... - P_N$, the high-level itinerary generated from this sequence would include the primitives and the trajectories of their region of existence $\mathbb{R}_e(P_1)$, $\mathbb{R}_e(P_2)$, to $\mathbb{R}_e(P_N)$. Each $\mathbb{R}_e$ contains 20 members if the primitive is generated from a periodic orbit, and 50 members if the primitive is obtained as an arc along a hyperbolic invariant manifold. The trajectories within each $\mathbb{R}_e$ are labeled U_i and form the nodes of a layer of the high-level itinerary, as in Figure 6.10b. This process is repeated for all N motion primitives composing the primitive sequence to obtain N layers. Nodes within the same layer are not connected to ensure that the final path geometrically resembles the motion primitive sequence generated in Step 3, thus leveraging the specific geometry represented by a primitive only once along the path. On the other hand, nodes in a layer are connected to all nodes in the following layer, ensuring flexibility in path search.

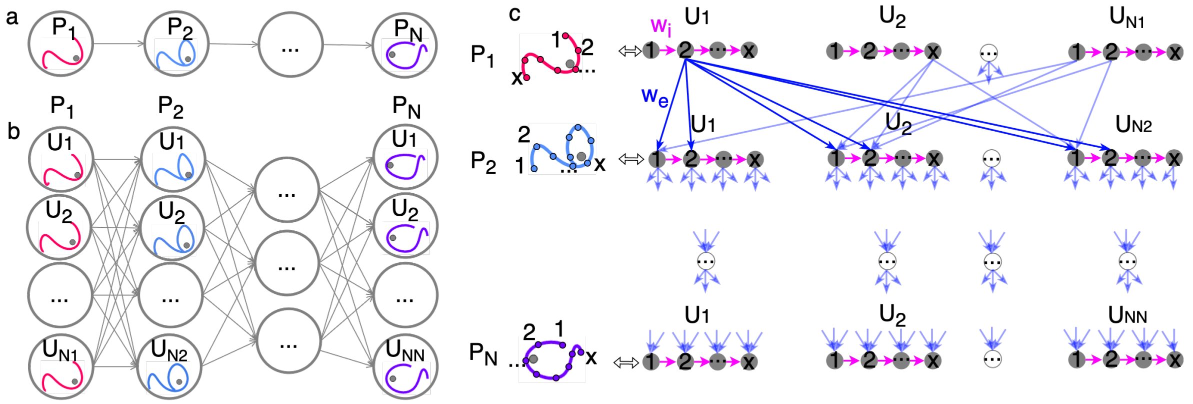


Figure 6.10: Constructing a refinement graph to support generating an initial guess from a motion primitive sequence

6.6.2 Constructing the Graph

The refinement graph is built using the states sampled along each trajectory as the nodes, consistent with the approach presented in Step 2. Each trajectory arc is sampled at maxima in curvature and at points equally spaced in arclength to create the sets of nodes consistent with the procedure outlined in Section 6.4.4. Considering all the U_m and U_p members of each manifold and periodic primitive set, the final graph will be composed of $N_{mp,m}U_mM_{s,m} + \sum_{i=1}^{N_{mp,p}} U_pM_{s,p,i}$ nodes.

The nodes representing the states along one trajectory are sequentially connected with quasi-zero-weight, set to $w_{i,i+1} = 10^{-15}$, to capture natural flow along these nodes. These are represented with magenta arrows and labeled w_i in Figure 6.10c. The nodes belonging to trajectories in the same sets are not connected as specified in the high-level graph. On the other hand, the nodes representing a state along one trajectory in the $\mathcal{R}_e(P_i)$th set are connected via directed edges to the composable nodes in the set $\mathcal{R}_e(P_{i+1})$, where the composability is defined in Step 2 considering the discontinuity in position, velocity direction and magnitude between samples states. Recall that the composability algorithm described in Section 6.4.2 outputs the indexes of the composable nodes between two primitive sets, as well as the index of the set members along which those nodes are composable. Therefore, consider the composable nodes between two primitive sets, labeled as $\mathcal{M}_{c,i}$ and $\mathcal{M}_{c,i+1}$. The edges between these nodes are created only between the U_jth and U_kth members of each set $\mathcal{R}_e(P_i)$th and $\mathcal{R}_e(P_{i+1})$ where the nodes $\mathcal{M}_{c,i}$ and $\mathcal{M}_{c,i+1}$ are composable.

As an example, consider the graph in Figure 6.10c. The second node along $\mathcal{R}_e(P_1)$ set is composable with the first node along the $\mathcal{R}_e(P_2)$. The composable nodes lies along U_1 and U_{N_1} of P_1 and U_1, U_2 and U_{N_2} of P_2. Therefore, edges are created between these nodes for the selected members of the primitive regions of existence. For the composable nodes, the edge weights are computed using the difference in the normalized discontinuity in position between two nodes, as expressed as

$$w_{i,j} = \frac{\Delta r_{i,j} - \Delta r_{min}}{\Delta r_{max} - \Delta r_{min}} \cdot [1 - 10^{-10}] + 10^{-10} \tag{6.13}$$

where $\Delta r_{i,j}$ is the distance between nodes a and b, Δr_{max} and Δr_{min} are the maximum and minimum distances between all the composable nodes within $\mathcal{R}(P_i)$ and $\mathcal{R}(P_{i+1})$. Normalizing the position difference between two nodes to within the range $w_{i,j} \in [1, 10^{-10}]$ mitigates biasing due to the size of the region of existence and the spacing between arcs sampled to span this region while ensuring that the edge weight remains positive. Figure 6.10c depicts these edges using blue arrows and labeled w_e.

6.6.3 Searching the Graph

The graph is searched using the A* algorithm [48] to obtain one path that minimizes the cumulative edge weights. The retrieved path is composed of the subset of nodes along the selected trajectory associated with each $\mathcal{R}(P_i)$, in the sequence obtained from Step 3. The cost of the path, which measures the total discontinuity of the initial guess, is given by the sum of edge weights connecting the selected nodes. 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 that each sequence is composed of at least two nodes along the same primitive or a member of its $\mathbb{R}_e$.

The initial guess construction process is visualized through the foundational transfer example 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 6.6. The unrefined sequence of primitives obtained from Step 3 is visualized in Figure 6.11a. In this figure, each motion primitive is displayed using uniquely-colored dashed lines, following the color scheme in Figure 7.1. These primitives exhibit discontinuities and overlapping regions. Following the refinement described in this section, the resulting initial guess is displayed in Figure 6.11b. Through this process, the final initial guess minimizes discontinuities and overlaps between arcs, such that the solution has a higher likelihood of successful corrections than the original sequence of primitives. This initial guess can be input to a corrections scheme.

An example of the variety of initial guesses obtained for the trajectory design in this section is presented in Figure 6.12. Subplots a to e show initial guesses for trajectories with two or three revolutions around Triton that remain in the cislunar region. Each trajectory is characterized by a diverse geometry, resulting from the sequential composition of a set of diverse primitives. Then, Figure 6.12f shows a trajectory initial guess that revolves around Triton a few times before leaving the cislunar region through the L_2 gateway, moving toward the L_3 point, to eventually reenter the L_2 gateway and reach the targeted L_2 Lyapunov orbit. Figure 6.12 illustrates the diversity of solutions generated from the motion primitives graph, as well as the effect of the refinement step in producing smoother and more continuous initial guesses.

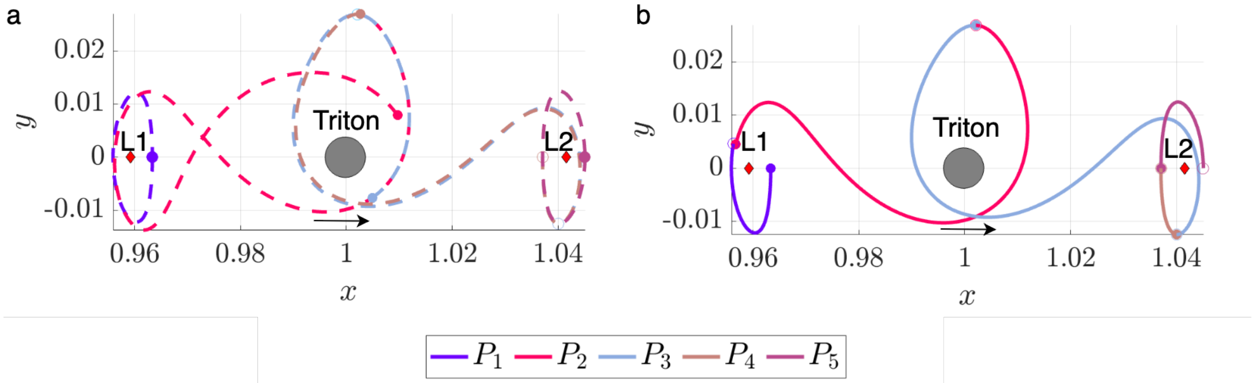


Figure 6.11: a) Coarse initial guess obtained from Step 3 and b) Refined initial guess obtained from Step 4

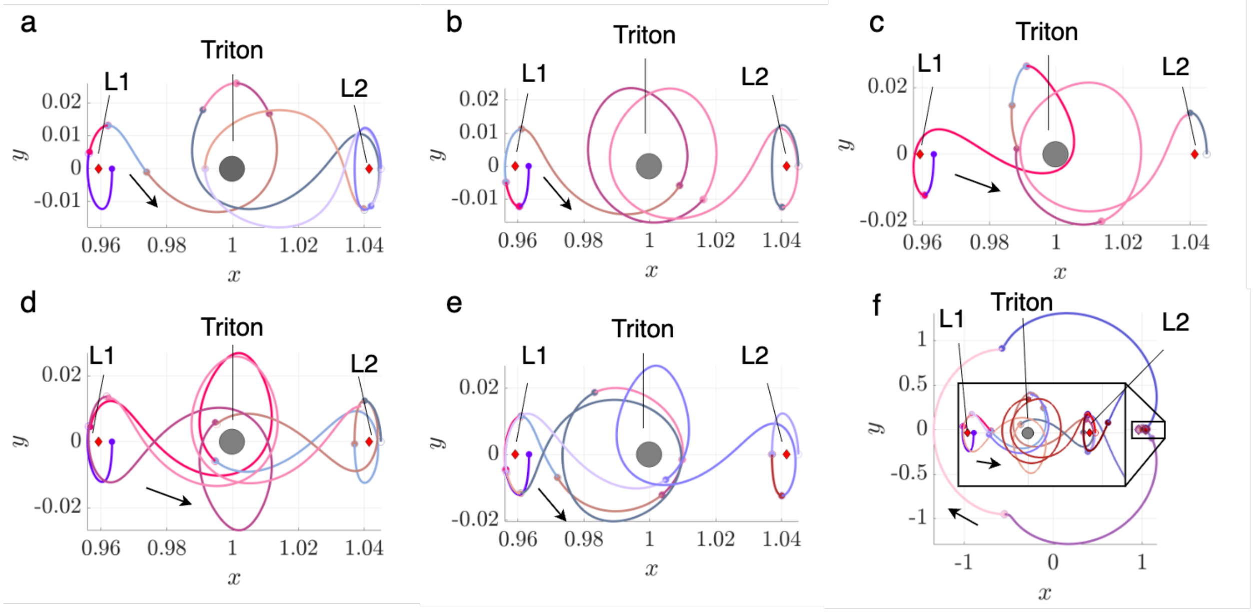


Figure 6.12: Selected examples of initial guesses for a transfer from an L_1 Lyapunov orbit at $C_J = 3.01400$ to an L_2 Lyapunov orbit at $C_J = 3.01377$ in the CR3BP

6.6.4 Summary of Procedure

Step 4 summarizes the process used to transform a primitive sequence into a trajectory initial guess. This process, referred to as initial guess refinement, utilizes a graph to transform a sequence of motion primitives into a set of nodes. Then, a search algorithm is used to find the sequences of nodes that minimize the sum of their position discontinuity. Overall, the refinement process creates a sequence of motion primitive arcs that minimize the discontinuity in position and the primitive overlap. A graphical, high-level overview of the technical approach described in this subsection is presented in the flowchart in Figure 6.13. This refinement step is governed by a few parameters and decisions, summarized in Table 6.4.

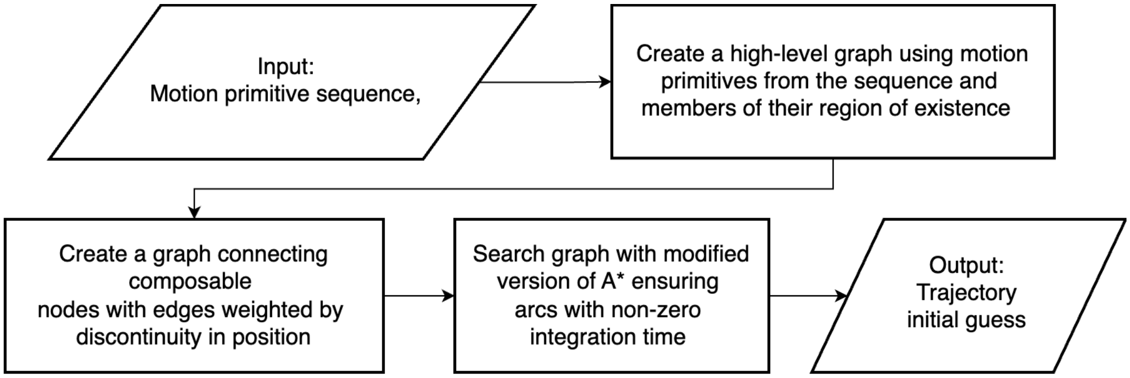

Input:
Motion primitive sequence,
Create a high-level graph using motion primitives from the sequence and members of their region of existence
Create a graph connecting composable nodes with edges weighted by discontinuity in position
Search graph with modified version of A* ensuring arcs with non-zero integration time
Output: Trajectory initial guess

Figure 6.13: Flowcharts of the process used to generate the trajectory initial guess from a motion primitive sequence

Governing parameter	**Influence**	**Value**
Starting and ending trajectory index	Indicate the member of the primitive's region of existence from which the path starts and ends	Index of the primitive trajectory.
Graph's edge weights	Influences which nodes are selected to obtain the minimum-cost path	Normalized discontinuity in position at composable nodes.
Search algorithm	Identifies the best sequence of nodes that minimizes the sum of the edge weights	Modified version of A* that ensures minimum arc size along each primitive.

Table 6.4: Table of governing parameters for constructing a motion primitive initial guess

6.7 Step 5: Trajectory Correction and Optimization

The trajectory initial guesses are corrected and optimized in the Neptune-Triton CR3BP. Then, the final continuous and optimal trajectories are also corrected in an ephemeris model that includes the gravitational influence of Neptune and its inner moons. This step leverages correction via collocation and constrained local optimization as presented by Smith and Bosanac [101].

6.7.1 Numerically Correcting Trajectories via Collocation

Coarsely designed initial guesses are corrected and optimized in the Neptune-Triton CR3BP and then in the ephemeris model by using collocation. Collocation relies on fitting polynomials of

Nth order to states distributed along arcs of a trajectory. These polynomials are constrained to 1) be continuous between arcs and 2) 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 polynomials are updated iteratively until the continuity and dynamical constraints are satisfied to within a specified tolerance. In this work, the formulation of the collocation problem follows the approach presented by Grebow and Pavlak [42]. This formulation was also used in the motion primitive approach to trajectory design developed by Smith and Bosanac [101]. This subsection presents a brief overview using states defined in the rotating frame for demonstrative purposes; however, the same formulation is straightforwardly extended to the inertial frame by using $\boldsymbol{X}_{B_i,3}$ rather than $\boldsymbol{x}$.

The first step of collocation involves creating a mesh of nodes. Each initial guess is discretized into n_s segments. Then, each segment is split into n_a arcs, and n_n nodes are placed along each arc. Within this mesh, the state at the kth node of the jth arc along the ith segment of the trajectory is described by the state vector $\boldsymbol{x}^i_{j,k}$. This segment is composed of n_a arcs, and Δt^i_j is the integration time along the jth arc. Although the number and spacing of segments and arcs are an input of the correction step, the number and location of nodes are defined by the order and type of the selected collocation scheme. Consistent with the approach presented by Grebow and Pavlak [42], this work uses a 7th order polynomial collocation scheme with a Legendre-Gauss-Lobatto (LGL) node spacing strategy. Thus, each arc is discretized into $n_n = 7$ nodes, placed at normalized times $\tau \in [-1, 1]$ that are equal to the roots of the derivative of the $(n_n - 1)$th order Legendre polynomial [23, 42, 111]. The odd-numbered nodes, i.e., $k = 1, 3, 5, 7$, are used to fit the polynomial $\boldsymbol{q}(\boldsymbol{x})$ along the arc and, therefore, are labeled as free nodes. On the other hand, the even-numbered nodes, i.e., $k = 2, 4, 6$, are used to assess the difference between the polynomial representation and the system dynamics; these nodes are labeled as defect nodes. Then, the states $\boldsymbol{x}^i_{j,k}$ and integration times Δt^i_j for each arc along the ith segment are used to generate a free variable vector $\boldsymbol{V}_i$, that is computed as

$$\boldsymbol{V}_i = \left[\begin{bmatrix} \boldsymbol{x}^i_{1,1} \\ \boldsymbol{x}^i_{1,3} \\ \boldsymbol{x}^i_{1,5} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{x}^i_{2,1} \\ \boldsymbol{x}^i_{2,3} \\ \boldsymbol{x}^i_{2,5} \end{bmatrix}^T \cdots \begin{bmatrix} \boldsymbol{x}^i_{n_a-1,1} \\ \boldsymbol{x}^i_{n_a-1,3} \\ \boldsymbol{x}^i_{n_a-1,5} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{x}^i_{n_a,1} \\ \boldsymbol{x}^i_{n_a,3} \\ \boldsymbol{x}^i_{n_a,5} \\ \boldsymbol{x}^i_{n_a,7} \end{bmatrix}^T \begin{bmatrix} \Delta t^i_1 \\ \Delta t^i_2 \\ \vdots \\ \Delta t^i_{n_a} \end{bmatrix}^T \right] \tag{6.14}$$

Note that $\boldsymbol{x}^i_{j,7}$ is not included in $\boldsymbol{V}_i$ because the last node of the jth arc coincides with the first node of the $(j+1)$th arc. The free variable vector for the entire trajectory $\boldsymbol{V}$ is then formed by the free variable vectors $\boldsymbol{V}_i$ for all n_s segments, resulting in a total of $((3n_n - 2)\sum_{i=1}^{n_s} n_a + 6n_s)$ 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 ith arc is defined as

$$\boldsymbol{F}^i_c = \begin{cases} (\boldsymbol{x}^{i+1}_{1,1} - \boldsymbol{x}^i_{n_a,n_n})^T \text{ if natural motion} \\ (\boldsymbol{r}^{i+1}_{1,1} - \boldsymbol{r}^i_{n_a,n_n})^T \text{ if impulsive maneuver applied} \end{cases} \tag{6.15}$$

for $i < n_s$. Then, the defect constraint vector enforcing the system dynamics at the defect nodes along the jth arc of the ith segment is defined as

$$\boldsymbol{F}^i_{d,j} = \begin{bmatrix} (\dot{\boldsymbol{q}}^i_{j,2}(\tau_2) - \dot{\boldsymbol{x}}^i_{j,2})\omega_2 \\ (\dot{\boldsymbol{q}}^i_{j,4}(\tau_4) - \dot{\boldsymbol{x}}^i_{j,4})\omega_4 \\ (\dot{\boldsymbol{q}}^i_{j,6}(\tau_6) - \dot{\boldsymbol{x}}^i_{j,6})\omega_6 \end{bmatrix}^T \tag{6.16}$$

where ω_k is the LGL weight associated with the kth collocation node, $\dot{\boldsymbol{q}}$ is the derivative of the polynomial along the arc with respect to normalized time τ, defined as $\tau = 2((t^i_{j,k} - t^i_{j,1})/\Delta t^i_j) - 1 \in [-1, 1]$, and $t^i_{j,k}$ is the time at the kth node along the jth arc and ith segment. In this expression, $\dot{\boldsymbol{x}}$ is the normalized time derivative of the state vector $\boldsymbol{x}^i_{j,k}$ calculated as

$$\dot{\boldsymbol{x}}^i_{j,k} = \frac{\Delta t^i_j}{2}\boldsymbol{g}(\boldsymbol{x}^i_{j,k}) \tag{6.17}$$

where $\boldsymbol{g} = [\dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]$. For all n_a arcs along the ith segment, the defect constraint vector is

$$\boldsymbol{F}^i_d = \left[\boldsymbol{F}^i_{d,1}, \boldsymbol{F}^i_{d,2}, \ldots, \boldsymbol{F}^i_{d,n_a}\right] \tag{6.18}$$

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

$$\boldsymbol{F}(\boldsymbol{V}) = \left[\boldsymbol{F}_c^1, \boldsymbol{F}_c^2, \dots, \boldsymbol{F}_c^{n_s-1}, \boldsymbol{F}_d^1, \boldsymbol{F}_d^2, \dots, \boldsymbol{F}_d^{n_s}\right]^T \tag{6.19}$$

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

Following the correction, the mesh that is composed of the 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 [92]. In this work, hybrid mesh refinement is implemented using the approach presented by Grebow and Pavlak and the method for error redistribution by Carl de Boor [42, 26, 95]; Smith and Bosanac previously implemented this approach for correcting primitive-based initial guesses [101].

The first step of mesh refinement is to redistribute the error in the polynomial approximation along the mesh. In this step, Carl de Boor's method is used to adjust the location of the nodes at the boundaries of each arc and, therefore, the integration time along each arc; the total number of arcs does not change. Once the boundary nodes of each arc have been updated, the polynomials computed from the previous mesh are used to place the LGL nodes along each arc in the new mesh. This error distribution is repeated until either: the maximum error difference along the current solution between any two arcs is $\leq 10^{-5}$; the maximum error difference along the current solution changed by $\leq$ 10% from the previous iteration; or a maximum number of iterations, selected in this work as 5, is exceeded. The values used to specify these termination conditions are the same as those used by Smith and Bosanac [101] but may be adjusted as needed.

Next, Control with Explicit Propagation (CEP) is used to iteratively merge or split arcs to balance reducing the dimension of the corrections problem with reducing the error along each arc [42, 92]. First, the state $\boldsymbol{x}_{j,1}^i$ at the first node along the jth arc within the ith segment is propagated forward in time for $\Delta t_j^i + \Delta t_{j+1}^i$. If the magnitude of the error between this propagated state and the state $\boldsymbol{x}^i_{j+1,n_n}$ (or $\boldsymbol{X}^i_{j+1,n_n}$) at the final node of the $(j+1)$th arc is below a tolerance of 10^{-13}, these two arcs are merged into a single arc. Next, the state $\boldsymbol{x}^i_{j,1}$ at the first node along the jth arc within the ith segment is propagated forward in time for Δt^i_j. If the error between this propagated state and the state $\boldsymbol{x}^i_{j,n_n}$ is above a tolerance of 10^{-12}, the arc is split into two arcs with the same integration time. For either split or merged arcs, the collocation nodes are recomputed between the updated boundary nodes using the polynomials from the previous mesh. The refined mesh is used in a new iteration of the correction process until there are no more arcs to merge or split, or a maximum of 10 iterations is exceeded [101].

6.7.2 Constrained Local Optimization

Continuous transfers obtained in the Neptune-Triton CR3BP via collocation are optimized to balance reducing the maneuver magnitude with geometrically resembling the initial guess. The optimal trajectory obtained from this step is then corrected in an ephemeris model using the same process, as detailed below. This optimization is performed using constrained local optimization via `fmincon` with the *sqp* algorithm in MATLAB [68, 84]. The free variable vector and the equality constraints vector used in these steps come from the collocation scheme described in the previous paragraph.

In this work, optimization is used to minimize a multi-objective cost function J, which balances minimizing geometric differences between the current trajectory and initial guess with minimizing the total Δv. The objective function is mathematically defined as

$$J = w_{geo}(\Delta \boldsymbol{r}_{ig-ct})^2 + w_{man}\sum_{i=1}^{n_m}(\Delta v_i)^2 \tag{6.20}$$

where $\Delta \boldsymbol{r}_{ig-ct}$ is the difference between the position vectors of each collocation node along the initial guess and current trajectory, and Δv_i is the magnitude of the ith of n_m impulsive maneuvers. The two competing objectives are balanced in J via two scalar weights, w_{geo} and w_{man}; these values are either set depending on the scenario or varied gradually in a continuation approach.

In addition to the equality constraints describing the collocation problem, additional path constraints can be included in this step. For an initial guess, the following inequality constraints can be used: maximum time of flight (TOF) as $TOF_{ig} \leq TOF_{max}$, maximum total Δv as $\Delta v_{TOT,ig} \leq \Delta v_{TOT,max}$, maximum or minimum single maneuver magnitude as $\Delta v_{i,min} \leq \Delta v_{i,ig} \leq \Delta v_{i,max}$, and the maximum or minimum distance from a body (Neptune B_1 or Triton B_2) as $d_{B_i,min} \leq d_{B_i,ig} \leq d_{B_i,max}$. These inequality constraints are appended to the constraint vector and additional rows are added to the Jacobian matrix.

The first two constraints are evaluated by summing the time or Δv at each free node along the trajectory, then subtracting the sum from the desired maximum or minimum value, obtaining a single entry to the non-linear constraint vector F_{maxTOF} or $F_{maxTotDv}$. For the third and fourth items, the constraints are evaluated at each free node (or at all free nodes where a maneuver is applied). The difference between the Δv or the distance from a body at that node and the corresponding minimum or maximum constraint value is then added to the vector of nonlinear constraints as F_{sDv} or F_{d_B}. The same approach is applied to the Jacobian of each constraint. If considering all the constraints, the non-linear constraint vector and its Jacobian will be computed as

$$\boldsymbol{F}_{nl} = [F_{sDv}^{i_M,j_M,1}; F_{d_B}^{i,j,k}; F_{maxTOF}; F_{maxTotDv}] \tag{6.21}$$

$$D\boldsymbol{F}_{nl} = [DF_{sDv}^{i_M,j_M,1}; DF_{d_B}^{i,j,k}; DF_{maxTOF}; DF_{maxTotDv}] \tag{6.22}$$

for $i = 1, ..., n_a$, $j = 1, ..., n_s$, $k = 1, ..., n_n$, where the subscript M indicates a segment and an arc where a maneuver is applied. The vectors of the linear and non-linear constraints, and their Jacobians, are merged and given to `fmincon` as the input for the *nonlcon* value, considering the linear constraints must satisfy $\boldsymbol{F}(\boldsymbol{V}) = 0$ and the non-linear constraints $\boldsymbol{F}_{nl}(\boldsymbol{V}) \leq 0$.

6.7.3 Correcting and Optimizing Motion Primitive Initial Guesses

In the first step of the correction process, each initial guess is discretized to form the initial mesh for the collocation. At this step, user-defined nodes are placed along the trajectory to create segments and arcs. In this work, segments are created between extrema in curvature, i.e., at maxima and minima in the curvature function described as in Section 6.3.1, while arcs are obtained by placing 7, 5, or 3 nodes at equal arclength distance. Then, seven collocation nodes are placed along each arc using the values of τ as described in Section 6.7.1. The set of collocation nodes for each arc along every segment forms the initial mesh. At this stage, maneuver locations can be placed at any node of the mesh and at the initial or final state of each primitive.

Using the collocation-based optimization scheme described in this section, an initial guess is corrected to minimize the maneuver costs while maintaining its geometry. Using continuation, the weights of the objective function are modified in a user-defined number of steps from $w_{geo} = 0.99$ and $w_{man} = 0.01$, to $w_{geo} = 0.01$ and $w_{man} = 0.99$, similar to the approach presented by Smith and Bosanac [101]. Note that $w_{geo} = 0$ and $w_{man} = 1$ are not used as the final values, as `fmincon` sometimes struggles to return a nearby solution within a reasonable computational time. At each step, the trajectory is corrected and optimized. This corrected trajectory seeds the initial guess for the next correction 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. At this step, the point-mass ephemeris dynamical model described in Section 2.3 is used to describe the gravitational influence of Neptune and its moons. In this work, the central body B_1 is Neptune, and the secondary bodies are the inner Neptunian moons: Triton, Naiad, Thalassa, Despina, Galatea, Larissa, and Proteus. Even though there are other Neptunian moons outside the orbit of Triton, these cannot be included in the correction step as information about these moons' mass is not directly available from the SPICE kernels [80]. The trajectories are corrected in the inertial frame with the collocation-based optimization described in the previous section, then they are visualized in the Neptune-Triton rotating frame and in the ICRF centered on Neptune when appropriate.

The optimization is obtained through MATLAB's fmincon function [68] with the sqp algorithm. For fmincon, the following options are specified: `OptimalityTolerance` $= 1e-3$, `StepTolerance` $= 1e-6$, and `ConstraintTolerance` $= 1e-12$. Moreover, the objective and constraint gradients are computed analytically and provided to the solver.

As an example, the top three trajectories in Figure 6.12 are first corrected and then optimized in the NT-CR3BP imposing a constraint of minimum altitude for Triton of $h_{min} = 300$ km, which is visualized in Figure 6.14 as a magenta circle around the moon. Each subplot shows the initial and final periodic orbits in blue, the optimal transfer in black, the initial guess in a gray dashed line, and the maneuver locations as red dots. Moreover, the final TOF, which is the time from the departure of the initial orbit to arrival into the target orbit, i.e., the time along the black trajectory, is listed on the bottom left corner of the subplots. Then, the sum of all the impulsive maneuvers is also shown in the bottom right corner. Throughout this work, impulsive maneuvers are allowed at maxima in curvature and at departure and arrival from the initial and final orbit, respectively. If during the correction or optimization problem, the maneuver magnitude becomes less than 0.01 m/s, then the maneuver is removed and the trajectory at that node is corrected for full state continuity.

Figures 6.14a, 6.14b show that the initial guesses obtained from Step 4 are close to the final corrected and optimal trajectories. In case of Figure 6.14c, the trajectory is morphed to meet the minimum altitude constraint with respect to Triton. However, the final trajectory still resembles the initial guess, especially in the departure arc from the L_1 Lyapunov and the arrival arc to the L_2 Lyapunov. The close resemblance of the final trajectories to the initial guesses shows that the primitives' composability and discontinuity minimization obtained through Steps 1 to 4 are the key to obtaining initial guesses that easily lead to low-cost trajectories.

The optimal trajectories in Figures 6.14a, 6.14b, and 6.14c are then corrected in the ephemeris model at the epoch of December 1st, 2045, at 00:00.000 UTC and visualized in the rotating frame in Figures 6.14d, 6.14e and 6.14f. In each transfer in this example and in the results presented in the next chapter, three additional revolutions are added around the initial and final orbits to anchor the transfer to the selected initial and final Lyapunov orbits. The time of flight below each figure indicates the number of days required for the transfer segment, i.e., from the maneuver to depart the L_1 Lyapunov orbit to the maneuver to insert the L_2 Lyapunov orbit, whereas the Δv value in the bottom right is the sum of all the impulsive maneuvers. By comparing the results in Figure 6.14, it can be noticed that the correction in the ephemeris maintains the geometry of the transfers, as well as the time of flight. The correction in a high-fidelity model influences the maneuver magnitude, making it slightly larger than the value in the CR3BP. This effect is expected as the ephemeris dynamical model includes gravitational forces that are not modeled in the NT-CR3BP, such as the effect of the smaller Neptunian moons. Overall, the optimal trajectories are easily corrected into a high-fidelity model, maintaining their original configuration and time of flight. This example demonstrates that initial guesses in a multi-body model can help recover diverse high-fidelity trajectories, especially in the proximity of the primaries.

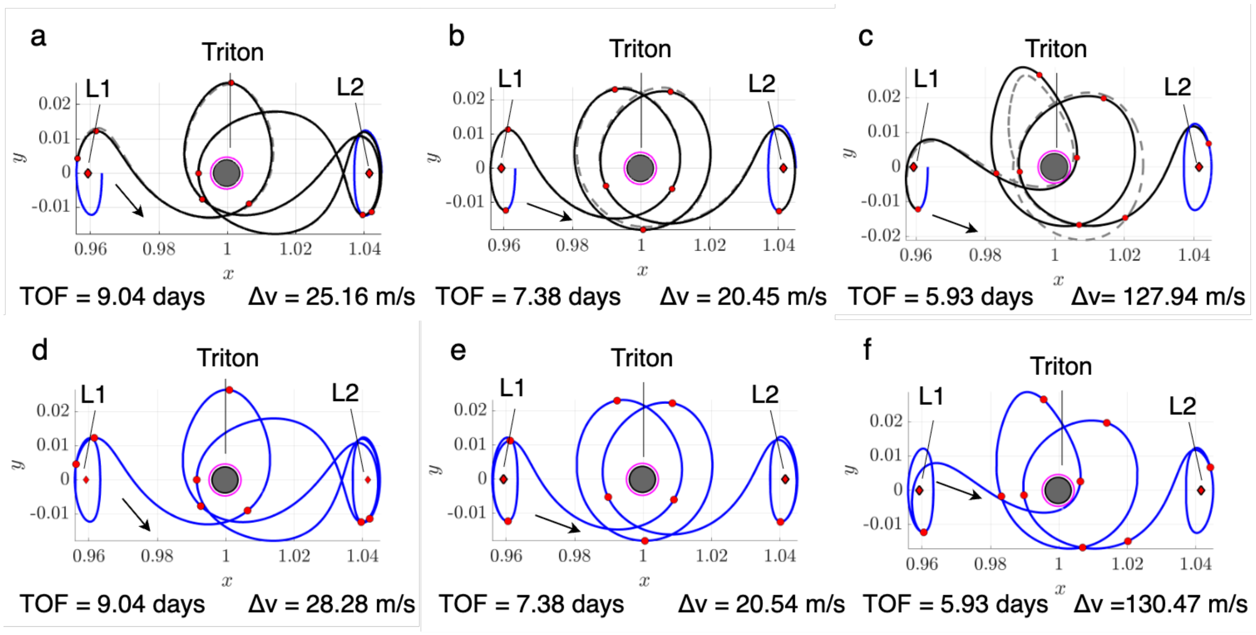


Figure 6.14: a), b) and c): Optimal trajectories in the NT-CR3BP model and d), e) and f) Optimal trajectories in the ephemeris model. All the trajectories are obtained from the first three initial guesses in Figure 6.12 for a transfer from an L_1 Lyapunov orbit at $C_J = 3.01400$ to an L_2 Lyapunov orbit at $C_J = 3.01377$ in the NT-CR3BP

6.7.4 Summary of Procedure

Step 5 describes the approach used to obtain continuous and optimal impulsive trajectories from the primitive-based initial guesses. A graphical, high-level overview of the technical approach described in this step is presented in the flowchart in Figure 6.15. Several parameters and decision variables employed in this step are summarized in Table 6.5.

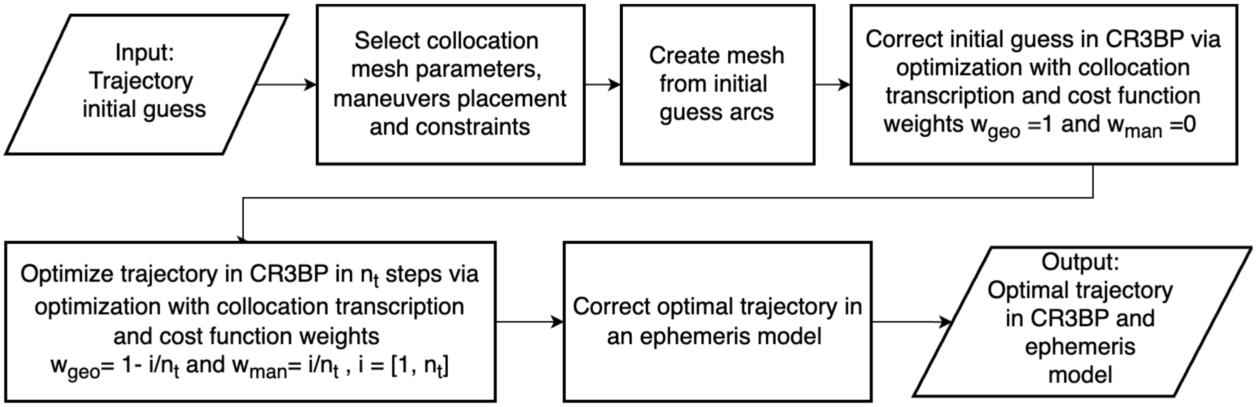

Input: Trajectory initial guess
Select collocation mesh parameters, maneuvers placement and constraints
Create mesh from initial guess arcs
Correct initial guess in CR3BP via optimization with collocation transcription and cost function weights w_{geo} =1 and w_{man} =0
Optimize trajectory in CR3BP in n_t steps via optimization with collocation transcription and cost function weights w_{geo}= 1- i/n_t and w_{man}= i/n_t , i = [1, n_t]
Correct optimal trajectory in an ephemeris model
Output: Optimal trajectory in CR3BP and ephemeris model

Figure 6.15: Flowcharts of the process used to generate optimal trajectory in an ephemeris model from a trajectory initial guess

Governing parameter	**Influence**	**Value**
Collocation mesh	Influences the speed and accuracy of the correction via collocation	Segments between maxima in curvature and arcs between points equally spaced in arclength.
Collocation polynomial order	Affects the accuracy of the correction	7th-order polynomial.
Collocation node spacing strategy	Identifies the location and the time components of the collocation nodes	Legendre-Gauss-Lobatto node spacing.
Maneuvers placement	Influences the cost and geometry of the final corrected trajectory	At maxima in curvature and between the initial and final orbits and the transfer trajectory.
Optimization cost function	Influences the properties of the final trajectory	Cost function as in Equation 6.20, with weights modified from $[w_{geo}, w_{man}] = [0.99, 0.01]$ for correction to $[w_{geo}, w_{man}] = [0.01, 0.99]$ for optimization.
Optimization algorithm	Influences the scale of the optimization problem to be solved and the solution of the optimization problem	Sequential Quadratic Programming (sqp) via MATLAB's fmincon function.
Ephemeris model	Identifies the dynamical model to be used for a trajectory's high-fidelity correction	Point mass gravity of Neptune and its main moons.

Table 6.5: Table of governing parameters for constructing the correction and optimization of the motion primitives trajectories

Chapter 7

Motion Primitive Approach to Designing Trajectories in the Neptunian System

The scenarios analyzed in this dissertation are categorized by the energy levels of the primitives employed in the transfer design. In particular, Section 7.1 presents a high-energy scenario, in which the motion primitive approach is used to compute transfers starting from the interplanetary arrival into the Neptunian system and ending with insertion into a science orbit. Then, a medium-energy case in Section 7.2 is used to study a trajectory design scenario for transferring from two resonant orbits with close flybys of Triton. Finally, Section 7.3 describes how the motion primitive approach is used to design trajectories in a low-energy regime to reach a low prograde orbit from a Neptune-centered orbit.

The motion primitive graph is then obtained starting from the definition of the high-level itinerary. The high-level itinerary is most often composed of a Start, a Transfer, and a Target set. Then, the motion primitive informed graph is constructed using samples along primitives as nodes and adding edges between the nodes that are considered composable.

Once the graph is completed, the custom k-best search algorithm is used to search for 50 diverse primitive sequences. These sequences can be composed of a variable number of primitives that minimize the overall discontinuity in velocity direction at the composable nodes.

Trajectory initial guesses are obtained from the motion primitive sequences after a refinement process. In this step, each motion primitive sequence is transformed into a graph where nodes are samples along the primitives and the trajectories in their $\mathcal{R}_e$, while edges connect the composable nodes. The graph is searched with a modified version of the A* algorithm to obtain the sequence of nodes along the primitive trajectories that minimizes the overall position discontinuity. The algorithm is modified to ensure that arcs along each primitive in the sequence have a minimum length, and no primitive is reduced to a single node. Initial guesses are always represented as a sequence of arcs following the color scheme in Figure 7.1.

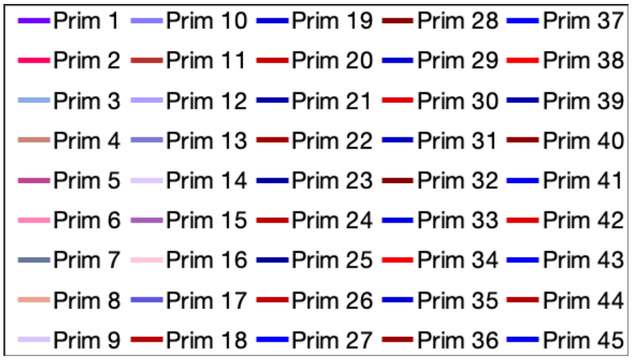


Figure 7.1: Color order for the primitive arcs composing the initial guesses

The initial guesses are corrected and optimized in the NT-CR3BP. The initial guess correction is formulated as a local constrained optimization problem with a collocation transcription. The collocation mesh is obtained by sampling the primitives at curvature extrema to create the mesh's arcs. Then, each arc is split into smaller segments obtained with points equally spaced in arclength. The number of points is chosen between $[3, 5, 7]$, and it is defined depending on the length of the arcs forming the mesh: for shorter arcs, a smaller number of segments is sufficient to represent the trajectory and obtain a corrected trajectory that resembles the initial guess; the opposite is true for longer arcs. Given this mesh, maneuvers are imposed at mesh nodes corresponding to maxima in curvature, and between the primitives connecting to the boundary conditions, i.e., between the first (or final) primitive and the consecutive (or previous) primitive.

The solutions for each design are analyzed by cost and geometric diversity and then used to define the trajectory trade space in a TOF and total Δv. Next, one or two trajectories per case are corrected in a point mass ephemeris model and visualized in an inertial frame centered on Neptune and in the rotating frame. The results in this chapter are computed by running the motion-primitive design approach with MATLAB R2022a [68] on a 2020 MacBook Pro with M1 chip and 8 GB of RAM.

7.1 High-Energy Transfer

This section illustrates the trajectory design to perform a Neptune Orbit Insertion (NOI) after arrival from an interplanetary trajectory. Following the insertion maneuver, the trajectory brings the spacecraft to a 1:7 resonant orbit, which is a potential science orbit in the Neptune-Triton CR3BP due to its favorable position with respect to Neptune's inner moons. The primitives leveraged for this design have a Jacobi constant in the range $C_J = [0.9, 2.5[$, corresponding to the highest energy primitives in the database.

7.1.1 Graph Construction

To construct the motion primitive graph, the high-level itinerary is first defined. In this case, the Start set of the high-level itinerary is composed of a trajectory arc obtained from the incoming leg of the interplanetary transfer from Earth. The arc is generated from a state at periapsis with respect to Neptune, computed by Dr. Reza Karimi during a prior study at NASA Jet Propulsion Laboratory. This state has the following characteristics [74]:

- Epoch at periapsis, t_1: October 2nd, 2045, 11:52:51 UTC
- Periapsis altitude relative to Neptune's surface: 2460.11 km
- Hyperbolic excess velocity, v_∞: 11.5252 km/s
- Declination angle, the angle between the velocity vector and the XY-plane of the Neptune-fixed frame labeled 'IAU_NEPTUNE' in the 'pck00011.tpc' kernel: $\delta = 8.3778°$

Since the original initial condition does not lie in the Neptune-Triton plane, the state components are modified to obtain a state with similar velocity, energy, and distance from Neptune that lies on the x-y plane of the NT-CR3BP rotating frame. This state is propagated backward and forward in time for 3.75 days to generate an arc that spans a vast region of the Neptunian system, offering a variety of maneuver locations [74].

The Target set of the high-level itinerary is composed of a 1:7 resonant orbit, characterized by a $C_J = 1.8$ and a period of 41.14 days. This orbit is selected as one of the Neptune-centered orbits with the lowest value of the Jacobi constant, thereby minimizing the energy discontinuity between the initial and final conditions of the transfer, while still providing relevant opportunities for scientific observations. This resonant orbit possesses a periapsis with Neptune with radius $r_{pN} = 108973.86$ km, allowing close observations of some of the inner moons of the planet, such as Proteus and Hippocamp, orbiting Neptune in semi-circular orbits with $a_P = 117,647$ km and $a_H = 105,300$ km, respectively [83].

Finally, the Transfer set is populated by the selected set of primitives with similar geometry and Jacobi constant to the initial and final primitives of the transfer. This set is composed of the primitives listed in the second column of Table 7.1.

Orbit Family	Selected Primitives
Res 1:x	PO $\pm x, +h$: 1:2, 1:3, 1:4, 1:5, 1:6
Res 2:x	PO $+x, +h$: 2:3, 2:5 and 2:7
Res 3:x	PO $+x, +h$: 3:5, 3:7 and PO $-x, +h$: 3:2
Res 4:x	PO $+x, +h$: 4:7 and Manifolds of PO $+x, +h$ 4:5 at $C_J = 2.3$

Table 7.1: List of primitives employed in the Transfer set the high-energy trajectory design scenario

Figure 7.2 provides a representation of the high-level itinerary: each set contains a plot of the selected primitives, while the arrows between sets and the symbols at the top of the set boxes describe the way edges are defined between and within each set. In the primitive subplots, each primitive is represented as a tick line with a unique color, and the shadowed region of the same color is the region spanned by the trajectories in the primitive's $\mathcal{R}_e$. The purple symbol at the top left corner of the Transfer set indicates that each primitive within a family is interconnected, i.e., a primitive of a family can be reached from any other primitive in the same family. Similarly, the black symbol on the top right corner of the Transfer set indicates that each pair of primitive families in the set is connected, i.e., any primitive in a family can be reached from any primitive of a different

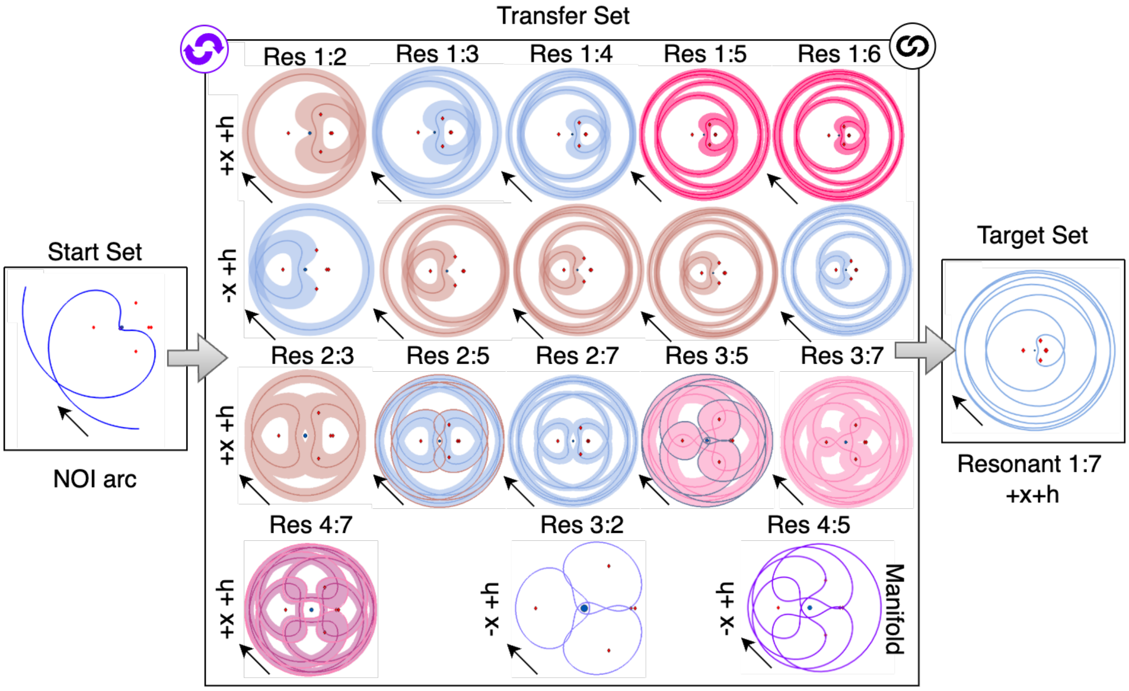


Figure 7.2: High-level itinerary for the high-energy trajectory design scenario

family. The motion primitive informed graph in this design is composed of 266 motion primitives, creating a graph with 3,871 nodes and 255,550 edges. On the specified computer, the graph is generated starting from the high-level itinerary in $\sim$ 29.33 minutes. Next, 50 motion primitive sequences are obtained from the graph using the k-best search algorithm in $\sim$ 59.08 minutes.

7.1.2 Initial Guesses

Initial guesses are obtained from the primitive sequences after the refinement process. In this case, the refinement was completed in $\sim$ 32.93 minutes on the specified computer. An example of 16 initial guesses obtained for the high-energy transfer design is presented in Figure 7.3. In the subplots from a to p, each primitive is represented by a unique color: the first primitive, i.e., the NOI arc, is always in purple, while the subsequent ones follow the color scheme in Figure 7.1.

The initial guesses in Figure 7.3 show a subsample of the solution space that is obtained with the motion primitive design approach. In particular, initial guesses have different geometries based on the primitive sequence that composes them. Primitives from the 4:5 resonant orbit manifolds produce initial guesses that remain within three nondimensional units from Neptune. On the other hand, primitives from the resonant periodic orbits, such as the 1:2 and 2:3 resonant orbits, form initial guesses with longer arcs at a larger distance from Neptune, which allow more rapid connections to the targeted 1:7 resonant orbit. An example of the effect of different primitives can be seen from the initial guesses in Figures 7.3a and 7.3m, compared to the ones in Figures 7.3c and 7.3h. The first are primarily composed of primitives obtained from the 4:5 resonant orbit manifolds, and thus are characterized by arcs that remain in the vicinity of the larger primary before inserting in the target 1:7 resonant orbit. In contrast, initial guesses that leverage the primitives from the periodic orbits, such as the ones in Figures 7.3c and 7.3h, are characterized by longer arcs that get further away from Neptune, following the geometries of each type of periodic orbit.

7.1.3 Final Transfers and Trade Space

The initial guesses are corrected and optimized in the NT-CR3BP. At this step, a constraint on the minimum altitude $h < 10$ km from Neptune is imposed on all the initial guesses to avoid collisions or excessive perturbations from the larger primary during maneuvers. Larger minimum altitude constraints or constraints on single or total Δv could be used in future work to incorporate specific mission requirements.

Figure 7.4 shows an example of the path obtained after correction and optimization. In particular, Figure 7.4a shows the trajectory initial guess, Figure 7.4b the corrected trajectory, and Figure 7.4c the final optimal transfer. The time of flight and total Δv are also added below the corrected and optimized trajectories to demonstrate the effect of the trajectory optimization. From the comparison of the orbits in Figure 7.4b and c, it is evident that the geometry of the initial guess is retained in the corrected and optimized trajectories, while the maneuver cost is reduced by approximately 14 times. However, more generally, the trajectory obtained after the optimization step is influenced by the discontinuity of the initial guess and by the choice of the maneuver placement.

The trajectories obtained after correction and optimization of the 16 initial guesses in Figure

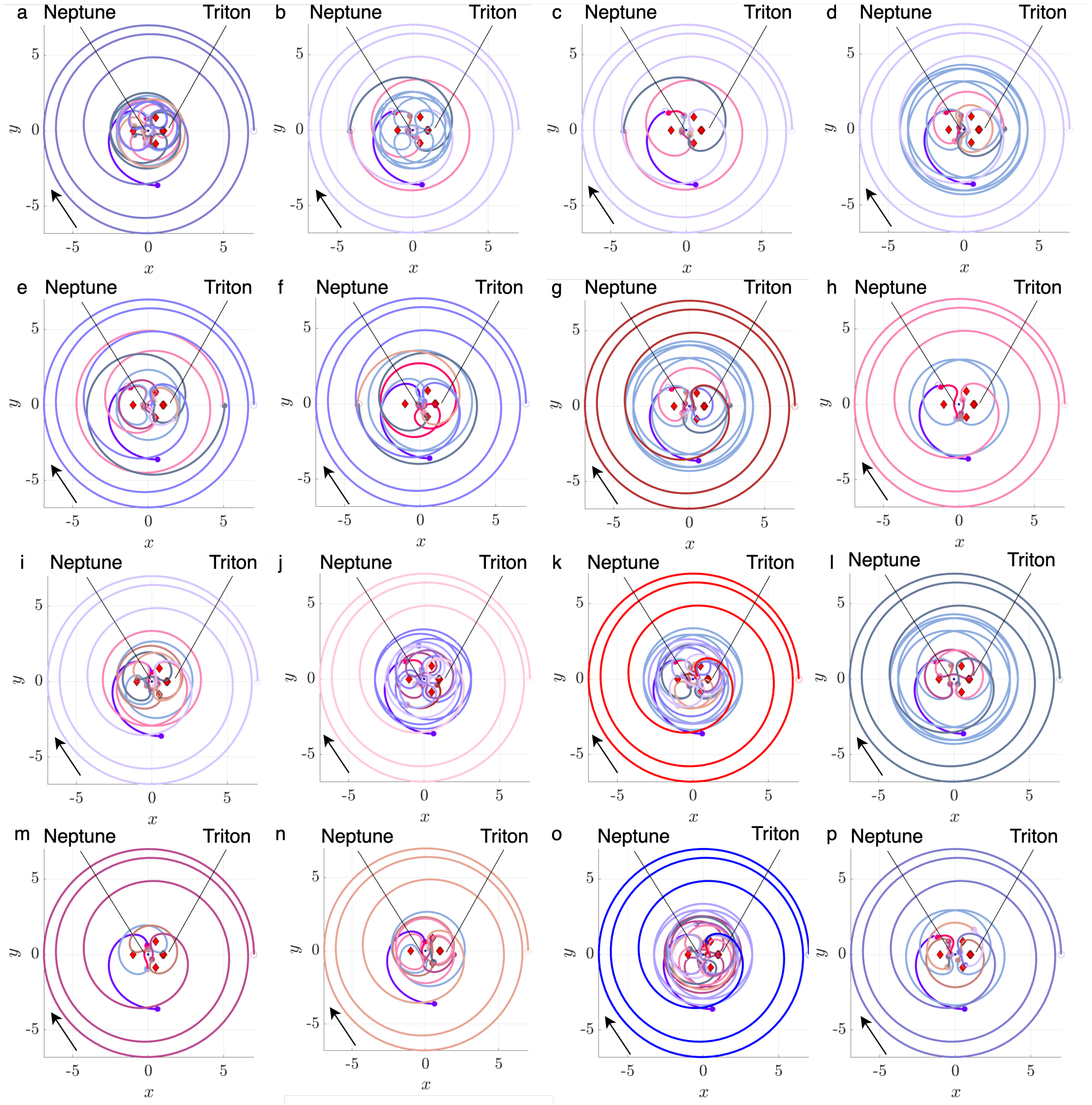


Figure 7.3: Selected initial guesses for high-energy trajectory design scenario

7.3 are presented in Figure 7.5. In each plot, the initial and final trajectories, i.e., the NOI arc and the 1:7 resonant orbit, are shown in blue, while the transfer is in black, and the initial guess is represented as a gray dashed line. The location of impulsive maneuvers is indicated by red circles, while a full blue circle and an empty blue circle represent the initial and final states of the trajectory,

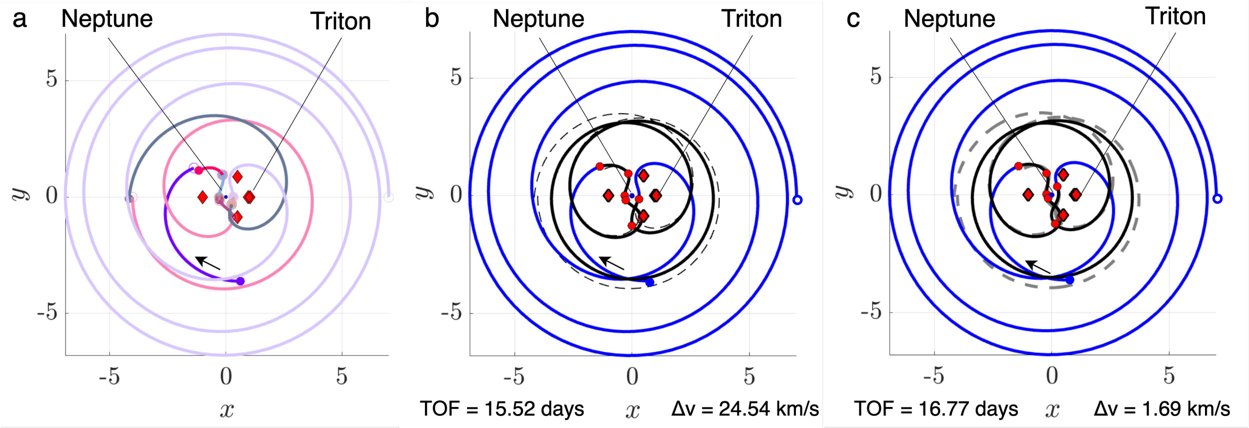


Figure 7.4: Example of a trajectory initial guess a) after correction b) and optimization c) with the associated TOF and Δv

respectively. Finally, the location of the equilibrium points is indicated with red diamonds. The primaries, scaled to possess their radii, are also highlighted with text and shown with blue and gray circles. The direction of motion is marked with a black arrow.

Final optimized transfers in Figure 7.5 closely resemble their respective initial guesses. This result demonstrates that the motion primitive design approach can obtain solutions that retain their geometry during correction, which may be important when designing trajectories to meet scientific objectives. The final optimal time of flight and total Δv values for all the 16 trajectories are presented in Table 7.1.3 and visualized in a TOF vs Δv tradespace in Figure 7.6. In this list, trajectories c and h have the lowest Δv, close to $\Delta v^* = 1.5$ km/s, the ideal value for the NOI maneuver [1] . After optimization, these two solutions are geometrically similar and approaching the shape of a 2:7 resonant orbit, thus indicating that further optimizations might result in a transfer along this periodic orbit.

The remaining trajectories in Figure 7.5 are characterized by longer times of flight and higher maneuver costs, depending on the initial guess. However, it should be noted that the choice of maneuver placement affects the overall maneuver cost, and therefore, the final cost can vary depending on the variations in maneuver locations. Further optimization and a different selection

[1] Private communication with Dr. Reza Karimi, NASA Jet Propulsion Laboratory, July 2023

of maneuver locations could be the venue of future work.

Initial Guess	TOF (days)	Δv_{tot} (km/s)
a	80.25	4.26
b	57.82	2.76
c	16.77	1.69
d	57.82	3.73
e	51.53	4.75
f	39.58	4.72
g	57.40	3.34
h	16.45	1.76
i	51.07	3.47
j	75.06	4.39
k	92.76	4.25
l	57.35	2.76
m	15.85	2.53
n	45.56	3.93
o	104.69	3.64
p	34.43	3.28

Table 7.2: Time of flight and total Δv for optimal transfers of the high-energy trajectory design case in Figure 7.5

Two of the most diverse transfers presented in this design are corrected in an ephemeris model and visualized in the inertial and rotating frames. These trajectories are corrected considering an initial epoch of September 28th, 2045, at 00:00.000 UTC, which is consistent with the start of the NOI arc, 3.75 days before the date at the periapsis state. The inertial frame representations are shown in Figure 7.7, where Figure 7.7a represents transfer in Figure 7.5c while Figure 7.7b represents transfer in Figure 7.5j. In these plots, the arc in gray is the arc along the NOI trajectory, and the remaining part of the trajectory is shown in black. The red circles are placed at the maneuver locations, while the purple circle is placed at the insertion into the target resonant orbit. The final Δv and TOF specified in the figure caption show that while the total duration of the transfer is preserved, the final maneuver requirement for the orbits corrected in the ephemeris model has increased compared to the trajectories in the CR3BP. This difference is expected, as the point-mass ephemeris model accounts for the gravitational effects of all interior moons orbiting near the periapsis of the trajectory, introducing additional perturbations to the motion.

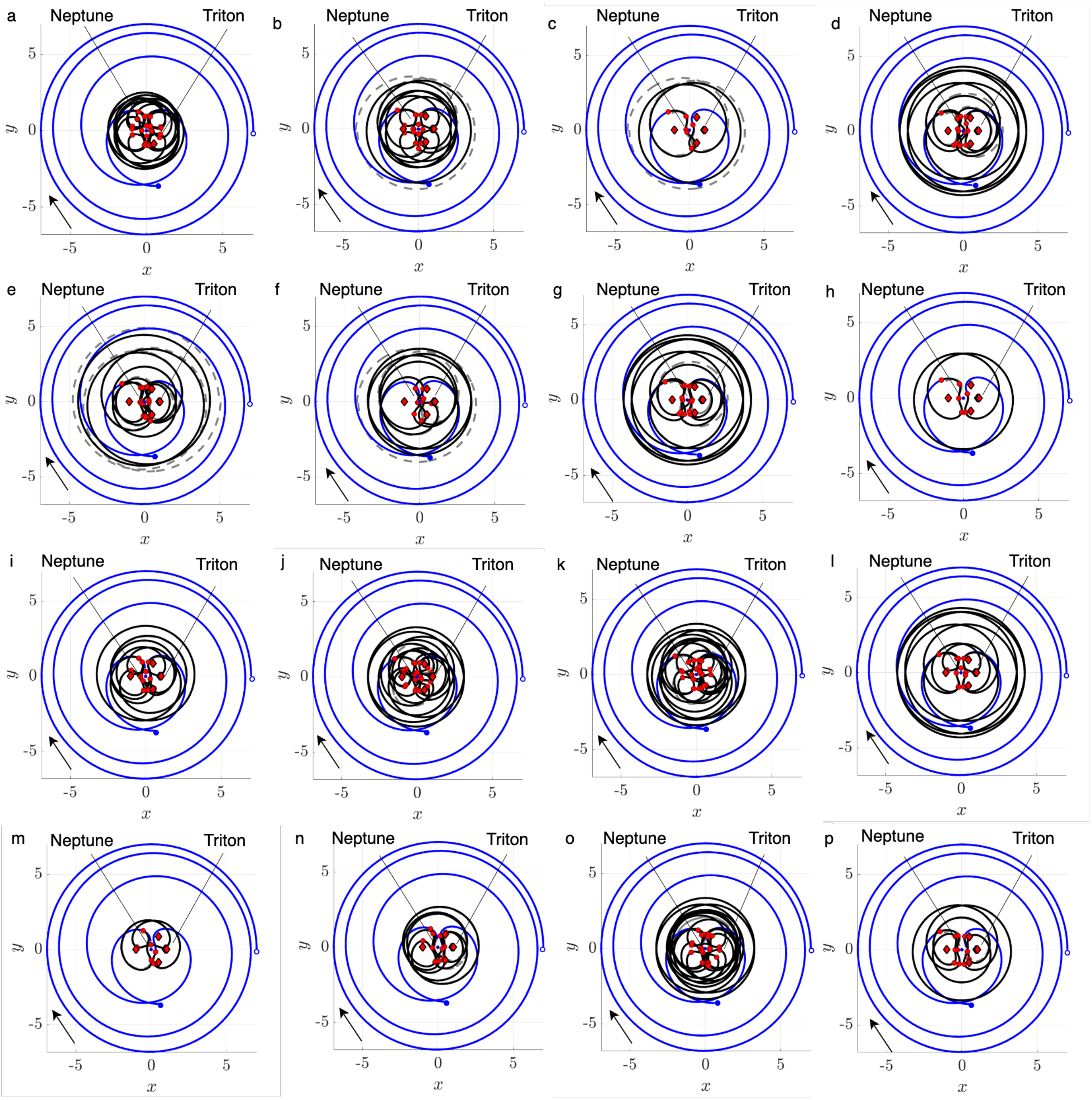


Figure 7.5: Final optimal trajectories for the high-energy trajectory design case

The same trajectories are also represented in the rotating frame in Figures 7.8a and 7.8b. The blue arcs are the trajectory segments along the initial and final orbits, while the red circles identify the maneuver locations. Comparing the trajectories in Figure 7.8 with the same trajectories corrected in the CR3BP in Figures 7.5c and 7.5j shows that the trajectories' geometry is preserved during the correction in a high-fidelity model.

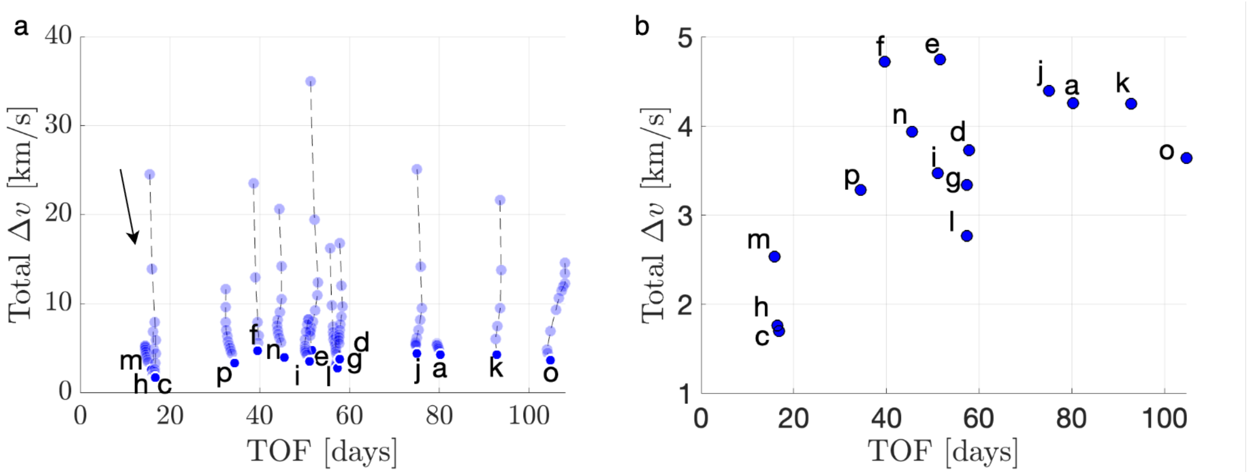


Figure 7.6: a) History of TOF and total Δv during the continuation steps to optimize each trajectory in Figure 7.5 and b) final cost in TOF and total Δv for the optimal trajectories in Figure 7.5

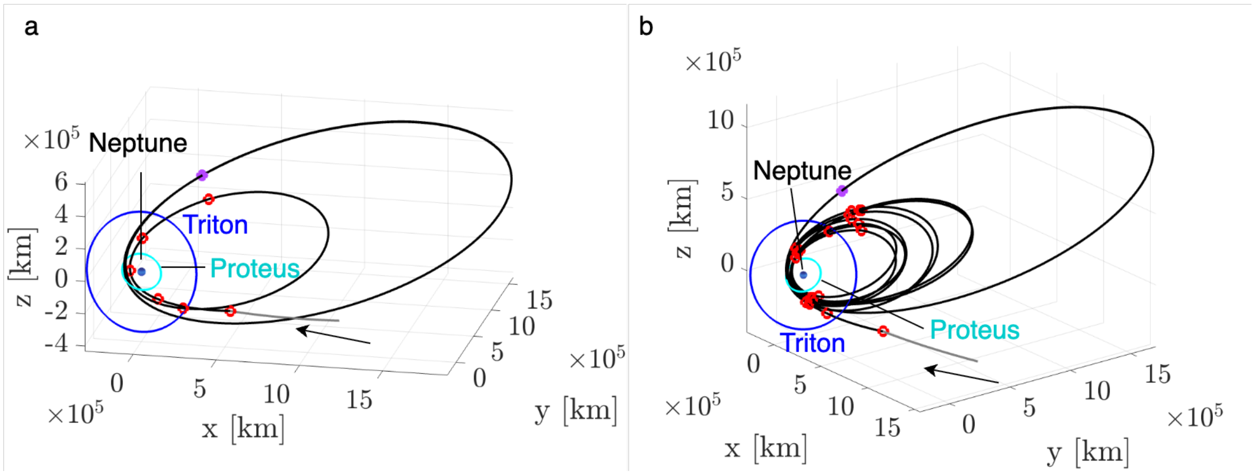


Figure 7.7: High-energy design trajectories corrected in an ephemeris model visualized in an ICRF about Neptune. Trajectory possesses: a) $\Delta v = 1.97$ km/s and TOF = 16.77 days and b)$\Delta v = 5.69$ km/s and TOF = 75.06 days

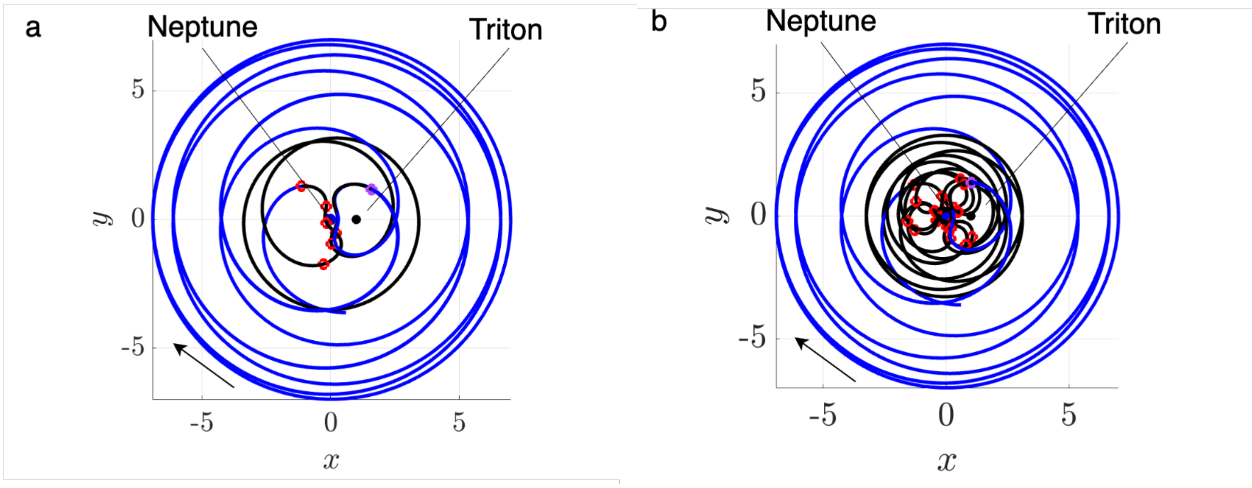


Figure 7.8: High-energy design trajectories corrected in an ephemeris model visualized in the rotating frame. Trajectory possesses: a) $\Delta v = 1.97$ km/s and TOF $= 16.77$ days and b) $\Delta v = 5.69$ km/s and TOF $= 75.06$ days

7.2 Medium-Energy Transfer

This section presents a trajectory design that leverages motion primitives in the Jacobi constant range $C_J = [2.5, 3.0[$. The initial and final orbits, a 4:3 resonant orbit and a 3:5 resonant orbit, are selected because they provide useful locations for close and slow flybys with Triton. At this energy level, several invariant manifolds and periodic orbits exist in the NT-CR3BP. The motion primitives associated with such solutions are leveraged to solve the trajectory design problem presented in this section.

7.2.1 Graph Construction

The motion primitive informed graph is obtained from the information of a high-level itinerary. The high-level itinerary is constructed by populating its Start, Target, and Transfer sets with selected motion primitives. In this design, the Start set contains a 4:3 resonant orbit with Jacobi constant of $C_J = 2.74$. This orbit has a period of 17.83 days and two Triton's periapsis passages per orbit period at an altitude of $h_p = 5699.72$ km. On the other hand, the Target set contains a $+x, +h$ 3:5 resonant orbit, characterized by $C_J = 2.99$, a period of 30.17 days. This orbit is selected as a potential science orbit for a Neptunian mission due to its two periapses with Triton at $h_p = 3180.76$ km within one orbit period. The initial and final orbits are represented in Figure 7.9 inside the Start and Target sets. Finally, the Transfer set consists of the motion primitives listed in Table 7.3.

Orbit Family	Selected Primitives
Res 1:x	PO $\pm x, +h$: 1:2, 1:3, 1:4, 1:5, 1:6, 1:7
Res 2:x	PO $+x, +h$: 2:3, 2:5 and 2:7
Res 3:x	PO $+x, +h$: 3:1, 3:2, 3:5, 3:7 and PO $-x, +h$: 3:7 Manifold of $+x, +h$ 3:5 at $C_J = 2.99$
Res 4:x	PO $+x, +h$: 4:5 and 4:7 and PO $-x, +h$: 4:7 Manifolds of PO $+x, +h$ 4:3 at $C_J = 2.74$

Table 7.3: List of primitives employed in the Transfer set of the medium-energy trajectory design scenario

The subplots in Figure 7.9 show the primitives selected from each of the periodic orbit families and invariant manifolds listed in Table 7.3. As for the previous case, the bold line within each subfigure is the primitive, while the shaded region represents the area spanned by the members of the primitive's region of existence. The purple and black symbols at the corners of the Transfer set box indicate that each primitive family is internally connected, and different primitive families are interconnected as discussed for the previous design case.

The graph obtained from this high-level itinerary comprises 237 primitives and is composed of 4779 nodes and 291,613 edges. The edges are constrained by limiting the connection to all the nodes that possess a velocity discontinuity $\Delta v < 3$ km/s. Then, the motion primitive graph is searched through the custom k-best search algorithm to find 50 primitive sequences. On the specified computer, the graph generation was completed in ~ 30.86 minutes, and the search of 50 primitive sequences was obtained in about 1 hour and 10 minutes.

7.2.2 Initial Guesses

Trajectory initial guesses are obtained from the motion primitive sequences after a refinement process, which was achieved in ~ 14.4 minutes on the specified computer. Figure 7.10 shows

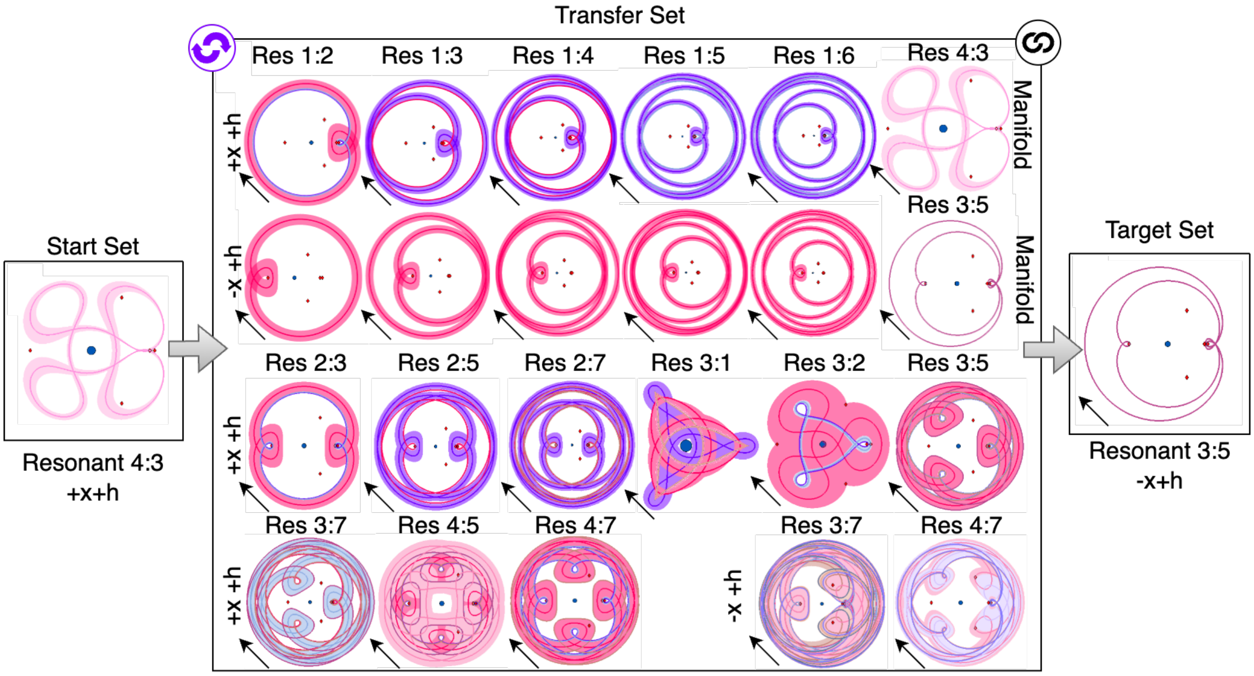


Figure 7.9: High-level itinerary for the medium-energy trajectory design case

a selection of 16 initial guesses for the medium-energy trajectory design scenario. Similarly to the results presented in the previous design case, each initial guess is composed of numerous different primitives, identified by different colors. Starting from the 4:3 resonant orbit in purple, the subsequent primitives are colored following the order outlined in Figure 7.1, the direction of motion is represented with a black arrow, and the primaries are labeled accordingly.

The solutions in Figure 7.10 highlight some of the geometries that can be obtained from concatenating the primitives selected for this design. Similar to the initial guesses obtained for the high-energy scenario, the solutions presented in this section generate a diverse solution space. In particular, solutions in Figure 7.10 are mainly composed of primitives generated from periodic orbits. However, arcs from the manifolds of the starting and target orbits are often used to facilitate departure from and insertion into the originating orbit, as they provide a lower-cost solution in terms of position and velocity discontinuity. The various combinations of primitives yield initial guesses with a range of geometries, maneuver cost, and time of flight, which helps explore the solution space.

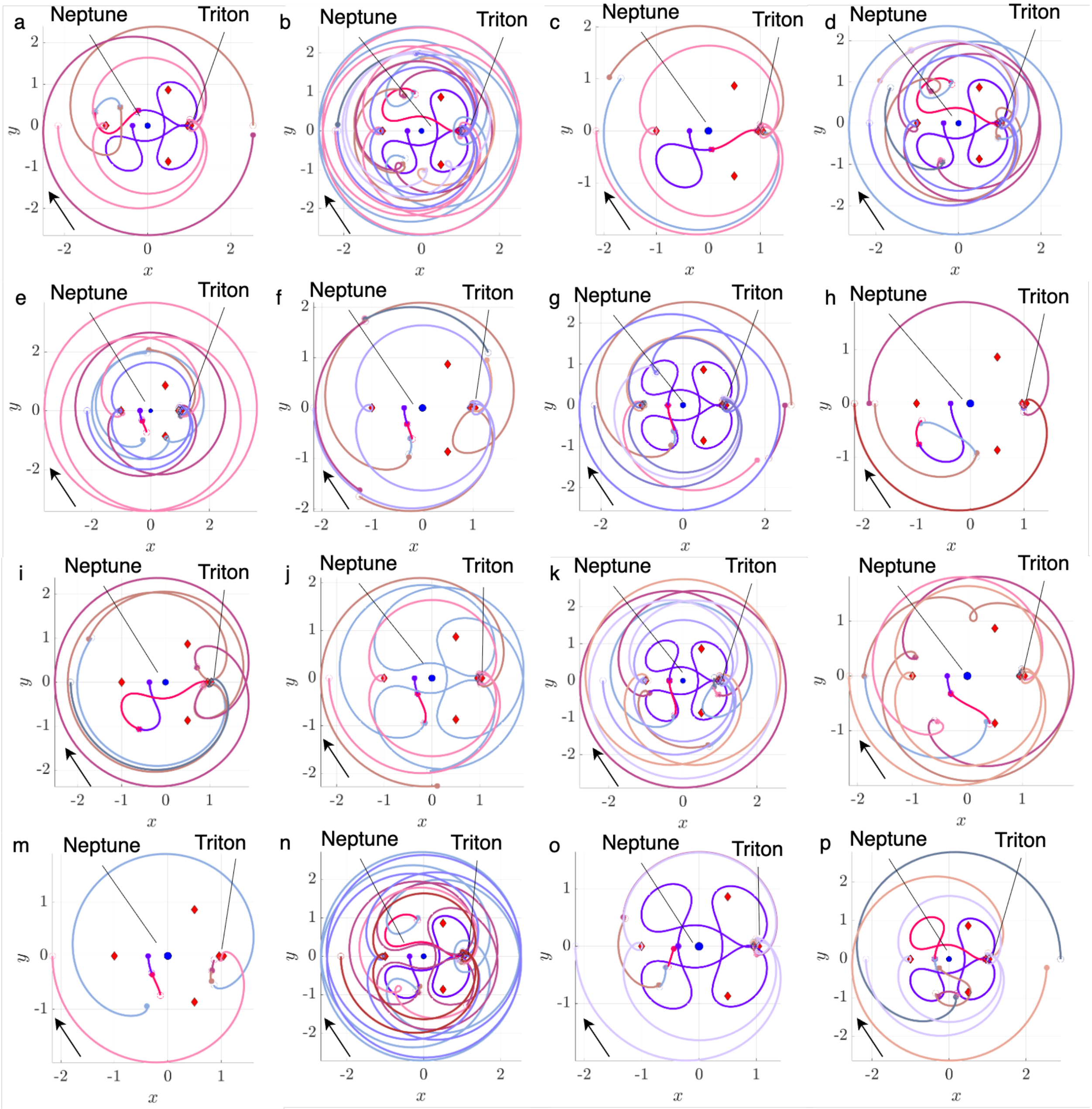


Figure 7.10: Selected initial guesses for medium-energy trajectory design scenario

7.2.3 Final Transfers and Trade Space

The initial guesses are corrected and optimized in the NT-CR3BP. The correction and optimization are obtained by imposing a constraint on the maximum single maneuver cost such that $\Delta v < 3$ km/s. The trajectories obtained after correction and optimization of the initial guesses in Figure 7.10 are presented in Figure 7.11. In each plot, the initial and final trajectories, i.e., the 4:3 resonant orbit and the 3:5 resonant orbit, are shown in blue, while the transfer is in black, and the initial guess is represented as a gray dashed line. The location of impulsive maneuvers is indicated by red circles, while a full blue circle and an empty blue circle represent the initial and final states of the trajectory, respectively. Finally, the location of the equilibrium points is indicated with red diamonds, the primaries, also highlighted with text, are shown with blue and gray circles, and the direction of motion is marked with a black arrow.

As seen in the high-energy trajectory design, the final optimal trajectories show a geometric resemblance to their initial guesses. However, in this trajectory design scenario, the final transfers have shifted from the initial guess to ensure state continuity. The larger discontinuities in the initial guesses for this design could be linked to two factors. First, the primitives selected for this design may span large regions of the position space but have small regions of overlap with other primitives. Secondly, when the initial guesses are composed of motion primitive arcs from different periodic orbit families, their sequence might present larger gaps in position space, as opposed to selecting primitives from the same family of hyperbolic invariant manifolds, such as the 4:5 manifold primitives from the previous design. Upon this observation, it can be concluded that the prevalence of a certain primitive family limits the geometric diversity of the initial guesses but increases their overall smoothness. In contrast, the diversity of primitive families leads to a more substantial diversity in final geometries, albeit with somewhat larger discontinuities in position. However, given the definition of composability between samples along primitives in Section 6.4.2, the maximum position discontinuity between two nodes is constrained by the width of the primitives' regions of existence at those nodes, thereby limiting the likelihood of producing highly discontinuous initial guesses.

The final optimal time of flight and total Δv values for all the 16 trajectories in Figure 7.11 are presented in Table 7.2.3. The majority of the corrected solutions are characterized by a variety of flight times and total maneuver costs, with trajectory h showing the minimum cost of $\Delta v_{tot} = 1.72$ km/s. Compared to the previous design case, the effect of the composability logic when including the limitation on Δv between two nodes helps produce cheaper initial guesses. Additionally, the least expensive trajectories represented in Figure 7.11h, 7.11o, 7.11l, and 7.11m are also among the paths with the shortest times of flight. In fact, these solutions leverage short arcs along some of the primitives originating from the highest resonance orbits, such as the 3:5, 4:5, and 4:7 resonant orbits, to immediately connect with the target orbit.

Initial Guess	TOF (days)	Δv_{tot} (km/s)
a	20.05	3.15
b	136.01	6.15
c	14.71	3.28
d	84.58	2.66
e	57.69	3.25
f	21.80	2.41
g	45.94	3.12
h	13.50	1.72
i	42.31	5.45
j	32.67	3.39
k	62.60	4.51
l	39.39	2.11
m	10.03	2.16
n	103.30	7.71
o	10.71	1.73
p	27.34	3.75

Table 7.4: Time of flight and total Δv for optimal transfers for the medium-energy trajectory design case in Figure 7.11

Trajectories in Figures 7.11d and 7.11h are corrected in an ephemeris model and visualized in the inertial and rotating frames. In this case, the trajectories are corrected considering an initial epoch of December 1st, 2045, at 00:00.000 UTC, which could be consistent with the start of the science phase following the orbit insertion and the first few phases of the mission. The trajectories

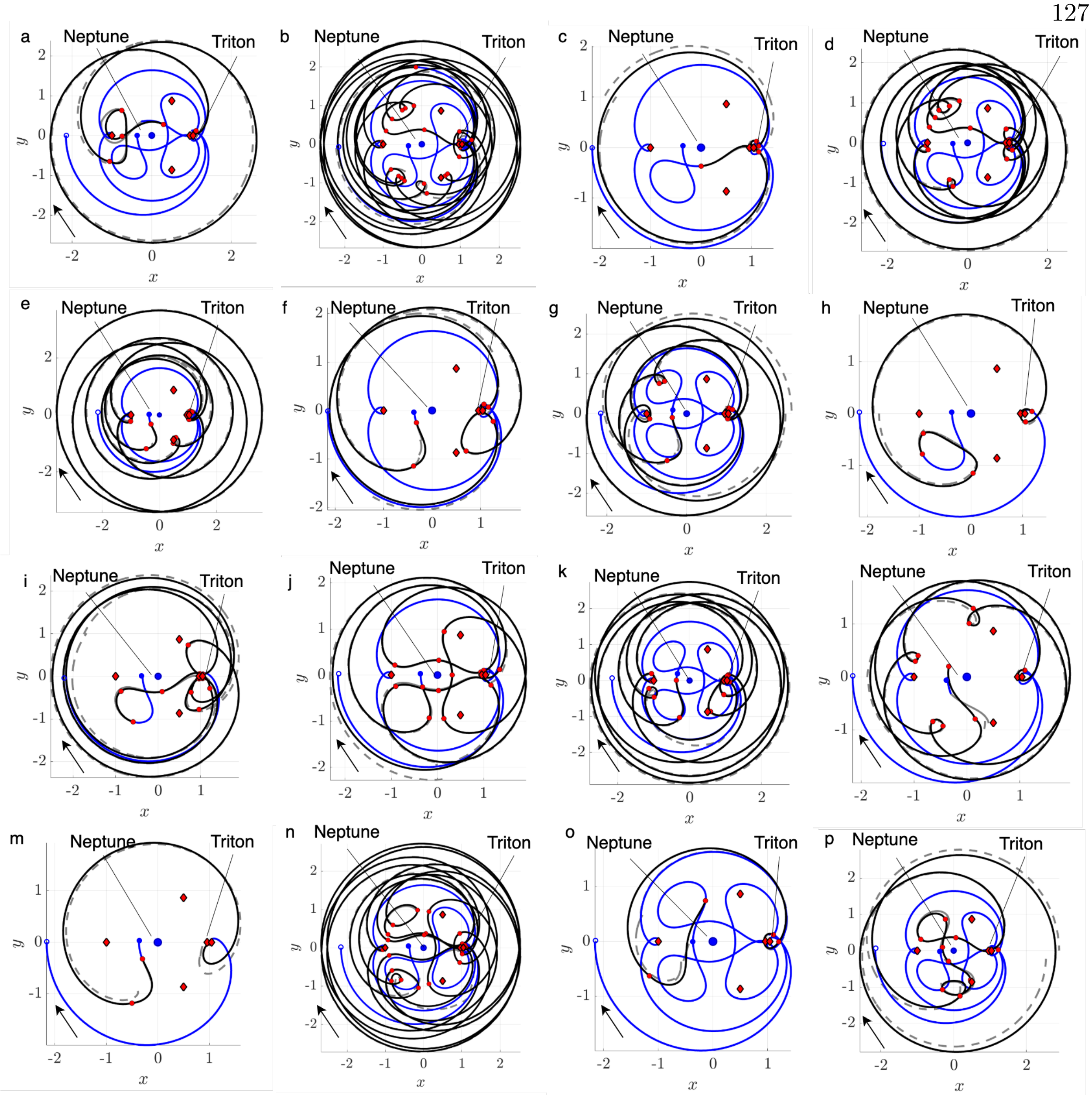


Figure 7.11: Final optimal trajectories for the medium-energy trajectory design case

in Figure 7.13 are shown in the ICRF frame centered on Neptune. The gray segment represents the starting arc of the trajectory, while the purple dot is the state at insertion into the target trajectory, and the red circles indicate the maneuver locations. The correction in the ephemeris model results in trajectories that preserve their geometry, proving that trajectory design in multi-body dynamics

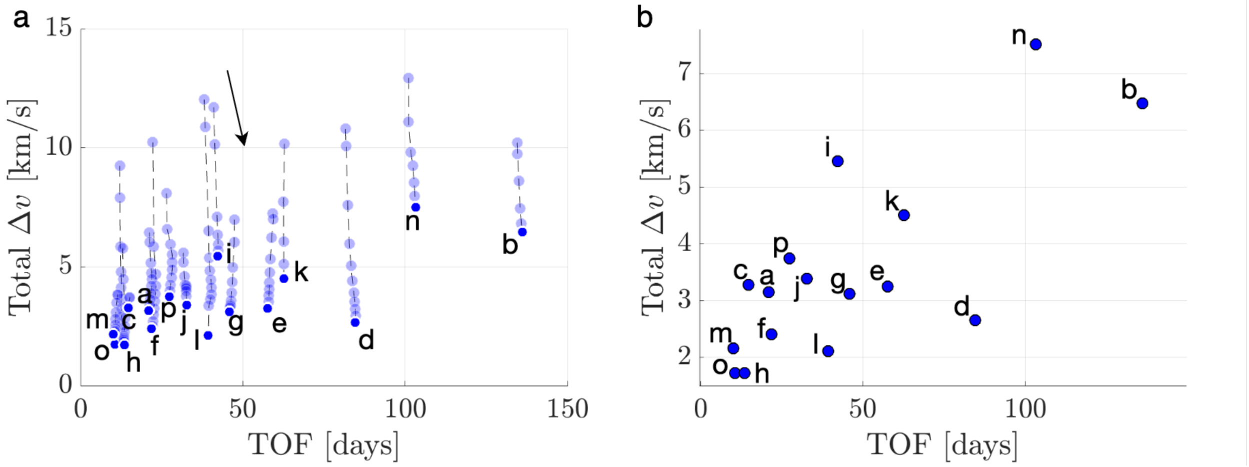


Figure 7.12: a) History of TOF and total Δv during the continuation steps to optimize each trajectory in Figure 7.11 b) Final TOF and total Δv for the trajectories in Figure 7.11

can help easily recover trajectories in a high-fidelity model. The final cost in TOF and Δv is higher than the trajectory in the NT-CR3BP, as expected, due to the additional gravitational influence of the inner moons of Neptune. The same trajectories are also represented in the rotating frame in Figures 7.14a and 7.14b, resembling the trajectories in Figures 7.11d and 7.11h. Similarly to the NT-CR3BP representation, the initial and final arcs of the trajectory are shown in blue, while the transfer is in black, and the maneuvers are shown with red circles.

7.3 Low-Energy Transfer

The low-energy scenario is used to design a transfer from a 3:2 resonant orbit to reach a low prograde orbit (LPO) around Triton, which could provide a strategic location for the scientific investigations of Triton and of the characteristics of the planetary system. This design leverages primitives from periodic orbits and invariant manifolds in the NT-CR3BP, possessing a Jacobi constant of 3.0 or higher.

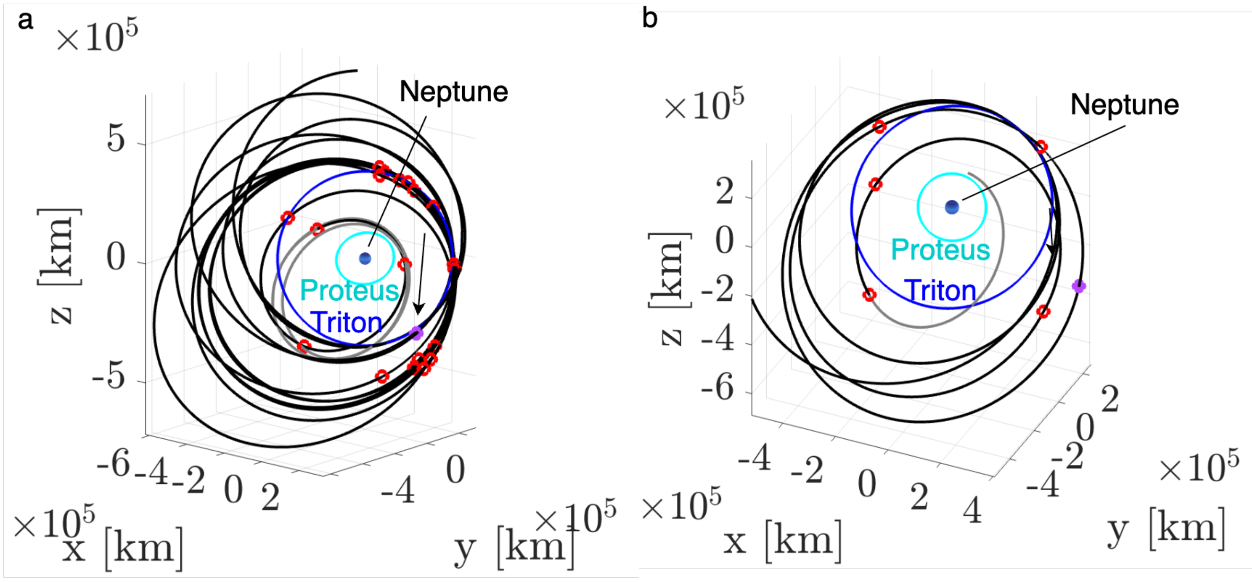


Figure 7.13: Medium-energy design trajectories corrected in an ephemeris model visualized in ICRF centered about Neptune. Transfer possesses: a) $\Delta v = 3.29$ km/s and TOF $= 84.58$ days and b) $\Delta v = 1.89$ km/s and TOF $= 13.50$ days

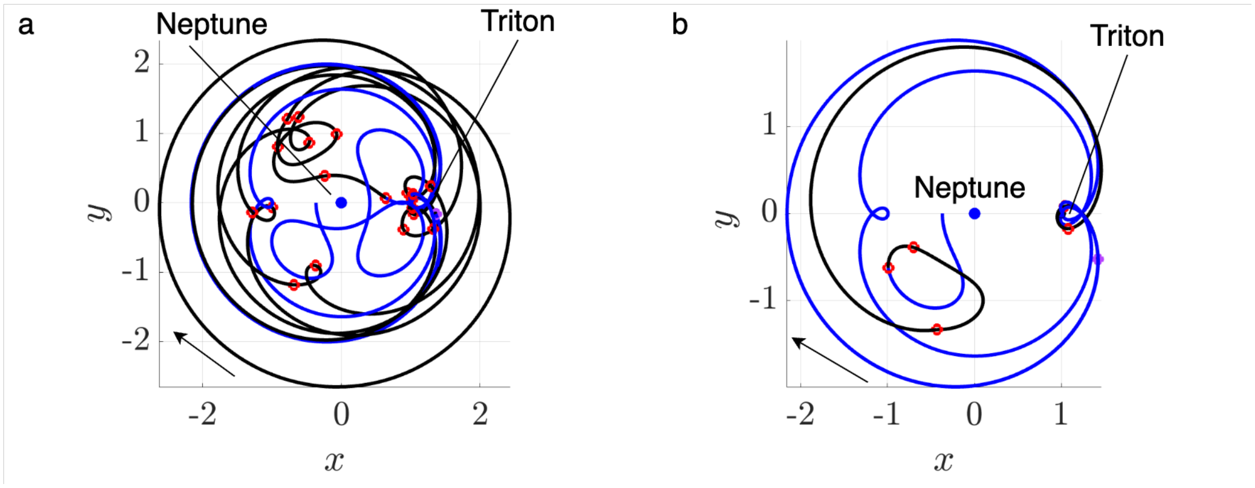


Figure 7.14: Medium-energy design trajectories corrected in an ephemeris model visualized in the rotating frame. Transfer possesses a) $\Delta v = 3.29$ km/s and TOF $= 84.58$ days and b) $\Delta v = 1.89$ km/s and TOF $= 13.50$ days

7.3.1 Graph Construction

In this trajectory design case, the high-level itinerary is defined with a Start set, two Transfer sets, and two Target sets, as represented in Figure 7.15. This high-level itinerary structure guides the design from the Neptune-centered orbit to the Triton-centered orbit in a more gradual manner, allowing the imposition of constraints over specific sets of primitives.

The Start set contains a 3:2 resonant orbit, characterized by a semi-major axis of $a = 407885.96$ km and a period of 11.74 days. This orbit provides opportunities for close observations of Neptune at the latitudes intercepted by Triton's orbital plane, inclined by 157.345° with respect to Neptune's equator. Therefore, an additional scientific phase could be planned along this trajectory before transitioning to the LPO.

The target orbit of this transfer is a low prograde orbit at $C_J = 3.0216$ with an orbital period of 11.850 hours and a minimum altitude of approximately $h_p = 2400$ km. These characteristics enable close observation of Triton while also mitigating the higher-level gravitational perturbations from the moon, which have yet to be mapped. Moreover, this orbit can be leveraged to leave Triton's proximity and explore other regions of the system, since its Jacobi constant is higher than the value at which the L_3 gateway closes.

Then, the Intermediate Target Set is composed of a L_2 Lyapunov orbit, which is selected to guide the trajectory from the Neptunian region to the L_1 and L_2 gateways. This orbit possesses stable and unstable manifolds that extend to the Neptunian region, spanning a vast portion of the phase space. This characteristic helps connect the starting resonant orbit and its neighboring periodic orbits with the orbital region around Triton.

The Transfer Set 1 contains primitives that connect the 3:2 resonant orbit with the L_2 Lyapunov. The starting 3:2 resonant orbit does not possess any invariant manifold; however, several natural structures from other families exist in its proximity and can be used to depart this orbit and approach the L_2 Lyapunov orbit. The complete list of primitives composing the Transfer Set 1 is reported in Table 7.5.

Orbit Family	Selected Primitives
Res 2:x	PO $+x, +h$: 2:1, 2:5 and PO $-x, +h$: 2:3, 2:5, 2:7
Res 3:x	PO $+x, +h$: 3:1, 3:5, 3:7 and PO $-x, +h$: 3:1, 3:2, 3:7
Res 4:x	PO $+x, +h$: 4:3, 4:5 and PO $-x, +h$: 4:3, 4:7
L_1 Lyapunov	Manifold of PO at $C_J = 3.0014$
L_2 Lyapunov	Manifold of PO at $C_J = 3.0037$
DPO	Manifold of PO at $C_J = 3.0139$

Table 7.5: List of primitives employed in the Transfer Set 1 of the low-energy trajectory design scenario

Some of the selected periodic orbits follow the same direction of motion as the initial and target orbits of this design, facilitating the recovery of a transfer following prograde motion. However, due to the retrograde direction of motion of the L_2 Lyapunov and its manifolds, additional retrograde orbits with resembling geometry and Jacobi constant are also added to the set.

Finally, the Transfer Set 2 is designed to obtain a transfer within the Triton orbital region, from the L_2 Lyapunov to the target LPO. This set is composed of the manifolds included in the Transfer Set 1 and listed in the last three rows of Table 7.5. Additionally, two path constraints are added in this set: first, primitives must remain within a radius of 0.1 from Triton, and second, primitives must have a periapsis above 10 km from Triton's surface. These constraints help retain only the primitives that exist within the orbital region of Triton and do not exit the gateways, while considering a minimum safe altitude from the moon to perform close flybys and maneuvers.

The high-level graph structure, composed of 1092 primitives, is used to form a motion primitive-informed graph that is composed of $10,126$ nodes and $838,869$ edges. The motion primitive-informed graph is generated in $\sim$ 3.13 hours, and 50 initial guesses are obtained using the k-best search algorithm over the graph in $\sim$ 2 days. Then, the motion primitives are refined to obtain the final initial guesses in $\sim$ 6 hours. The longer runtime compared to the previous scenarios is due to the graph's larger size, the longer primitive sequences, and the limited computational power of the specified computer. Improving the efficiency and reducing the memory requirements for searching larger graphs can be an interesting avenue for future work.

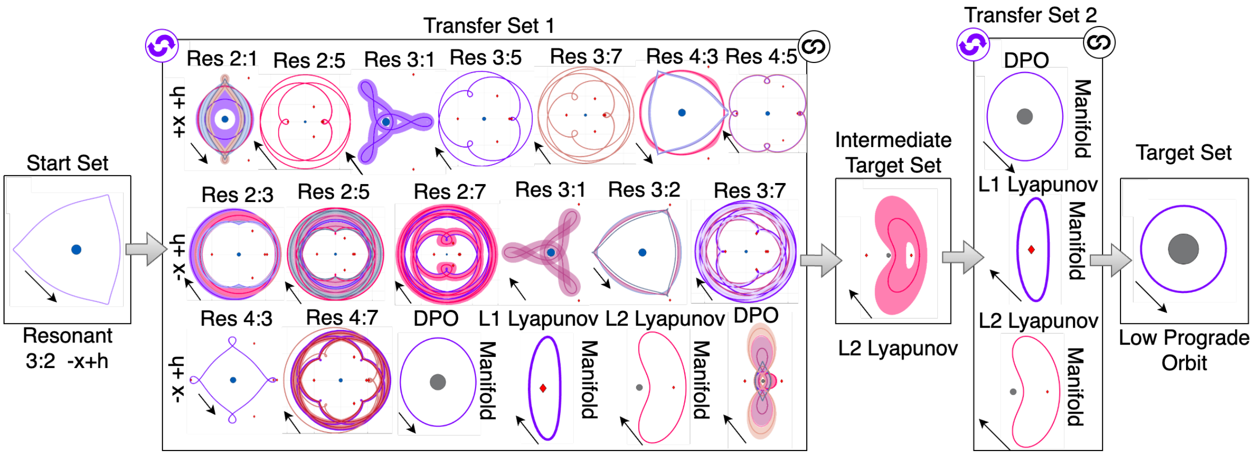


Figure 7.15: High-level itinerary for the low-energy trajectory design

7.3.2 Initial Guesses

The refined motion primitive sequences form the trajectory initial guesses. A subsample of these initial guesses is presented in Figure 7.16, where two views are used to capture the global geometries of the trajectories. In the first and third columns, 16 selected initial guesses are plotted in a Neptune-centered view that allows for analysis of the departure from the 3:2 resonant orbit in purple. The second and fourth columns of Figure 7.16 present a zoomed-in view of the region around Triton for analysis of the approach to the LPO. Each pair of subplots represents one initial guess, labeled with letters from a to p. The color scheme in the initial guesses plots indicates the order in which the motion primitive arcs are composed to create the initial guess and follows the order listed in Figure 7.1.

The selected initial guesses show diverse geometries, especially in the region near Triton. The majority of the initial guesses have a smooth sequence of primitives in position space, except for paths a, j, and n, where the primitives connecting to the L_2 Lyapunov have a position difference of ~ 0.025 nondimensional units. However, this position gap can be addressed during correction, as a low level of discontinuity characterizes the majority of the initial guess.

The geometries of the initial guesses in Figure 7.16 show that the L_2 Lyapunov is not always leveraged in its integrity to reach the LPO. Solutions such as the ones in Figure 7.16c and 7.16k leverage only small arcs along the periodic orbits before inserting into the primitives that connect to the LPO. These results demonstrate that the intermediate target effectively serves as a bridge between the two regions of the system. Moreover, it can be observed that the selection of the L_2 Lyapunov orbit as the intermediate target does not bound the transfers to approach the Triton region from the L_2 gateway. On the contrary, the manifolds selected in the Transfer Set 2 provide solutions that cross both L_1 and L_2 gateways while still connecting to the selected Lyapunov orbit, creating a variety of approaching solutions.

7.3.3 Final Transfers and Trade Space

The initial guesses are corrected and optimized within the NT-CR3BP framework. At this stage, a constraint is imposed to ensure the minimum altitude remains above 10 km from Neptune, thereby avoiding trajectories that impact the smaller primary. The trajectories obtained after correction and optimization are presented in Figures 7.17. In each plot, the initial and final trajectories, i.e., the 3:2 resonant orbit and the LPO, are shown in blue, while the transfer is in black, and the initial guess is represented as a gray dashed line. The maneuvers are represented as red circles, while a full blue circle and an empty blue circle represent the initial and final locations of the transfer, respectively. Finally, the cost in time of flight and total Δv associated with each trajectory is listed in Table 7.3.3 and in the trade spaces in Figure 7.18.

The optimal trajectories in Figure 7.17 closely resemble their respective initial guesses, especially after departure from the 3:2 resonant orbit. However, some optimal solutions exhibit a more significant evolution from the initial guesses in the vicinity of the Triton orbital region. This difference is expected in the trajectories with larger discontinuities in the initial guesses, such as the ones in Figure 7.17a, 7.17j, and 7.17n. However, even some of the most continuous paths, like the ones in Figure 7.17c, 7.17e, and 7.17o, evolve during optimization in the search for low-cost maneuver cost trajectories. Many of the primitives approaching the LPO eventually evolve into DPO-like orbits, characterized by the diamond shape about the second primary. The choice of maneuver placement could also drive the diversity between the initial guess and the final orbit.

Overall, the solutions in Figure 7.17 provide feasible transfers from a Neptune-centered orbit to an LPO. The majority of the solutions depart the initial orbit with retrograde motion and only change it back to prograde motion when reaching the LPO. Trajectory in Figure 7.17g is the only prograde transfer in this solution subsample. This trajectory is also the path with the lowest associated maneuver cost of $\Delta v = 0.91$ km/s, possibly indicating that a change in direction of motion might increase the overall maneuver requirement.

This trajectory is corrected in an ephemeris model at the initial epoch of December 1st, 2045, at 00:00.000 UTC. The trajectory is visualized in a rotating and an inertial frame in Figure 7.19. The representation in the inertial frame in Figure 7.19c shows the approach from the Neptune-centered orbit in the most inner part of the trajectory, to the orbit of Triton in blue, until the insertion into the LPO. The representation of the same orbit in the rotating frame in Figure 7.19a and 7.19b demonstrates that the geometry of the trajectory designed in the CR3BP is maintained after correction in the ephemeris model.

Initial Guess	TOF (days)	Δv_{tot} (km/s)
a	71.48	4.58
b	27.09	1.65
c	56.50	2.62
d	31.19	2.14
e	46.88	2.52
f	36.22	3.67
g	18.42	0.91
h	72.57	4.24
i	42.85	2.17
j	27.28	1.60
k	34.45	2.96
l	50.65	3.66
m	46.47	1.95
n	61.79	2.64
o	18.22	1.88
p	27.22	1.42

Table 7.6: Time of flight and total Δv for optimal transfers for the low-energy trajectory design case in Figure 7.17

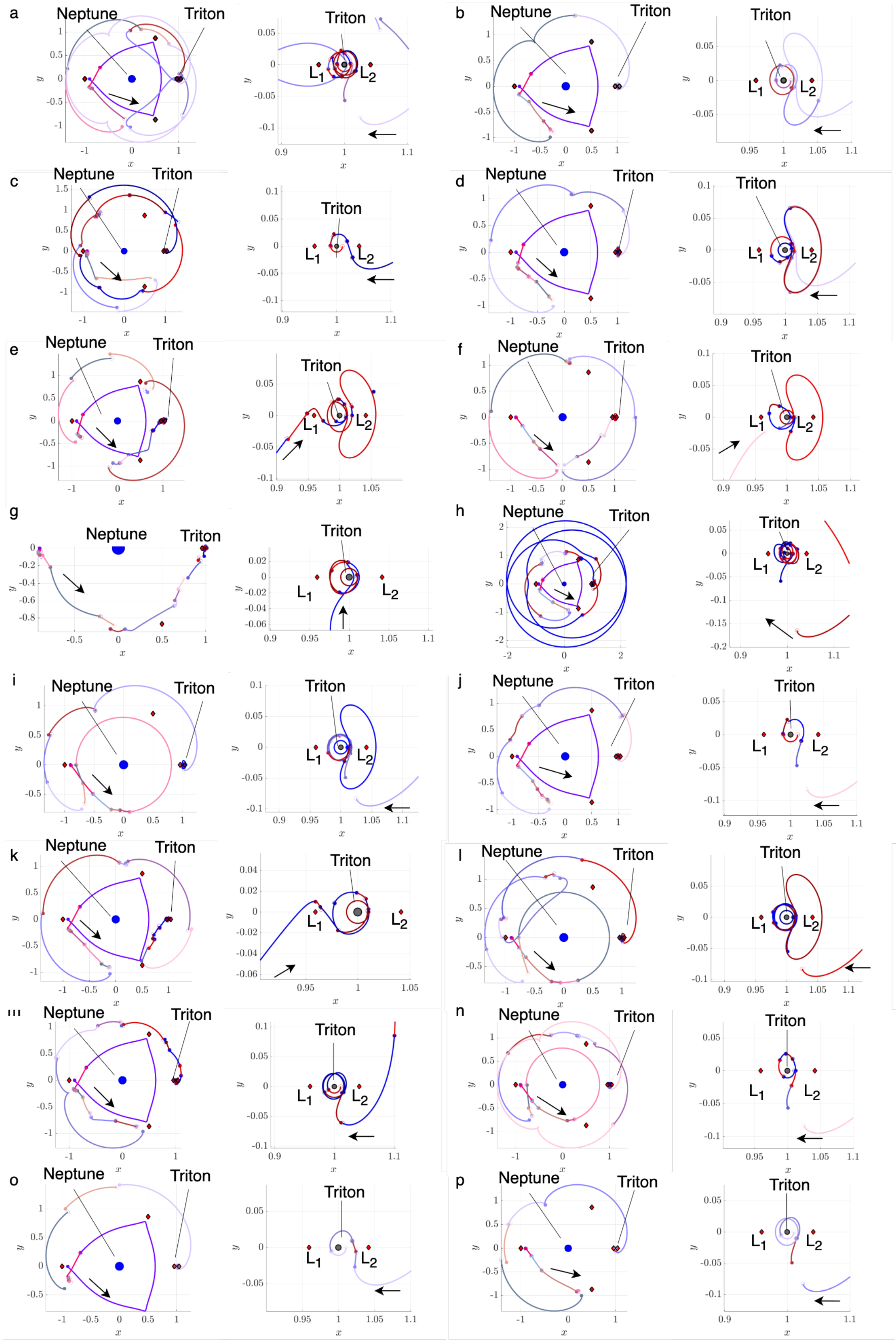


Figure 7.16: Selected initial guesses for low-energy trajectory design scenario

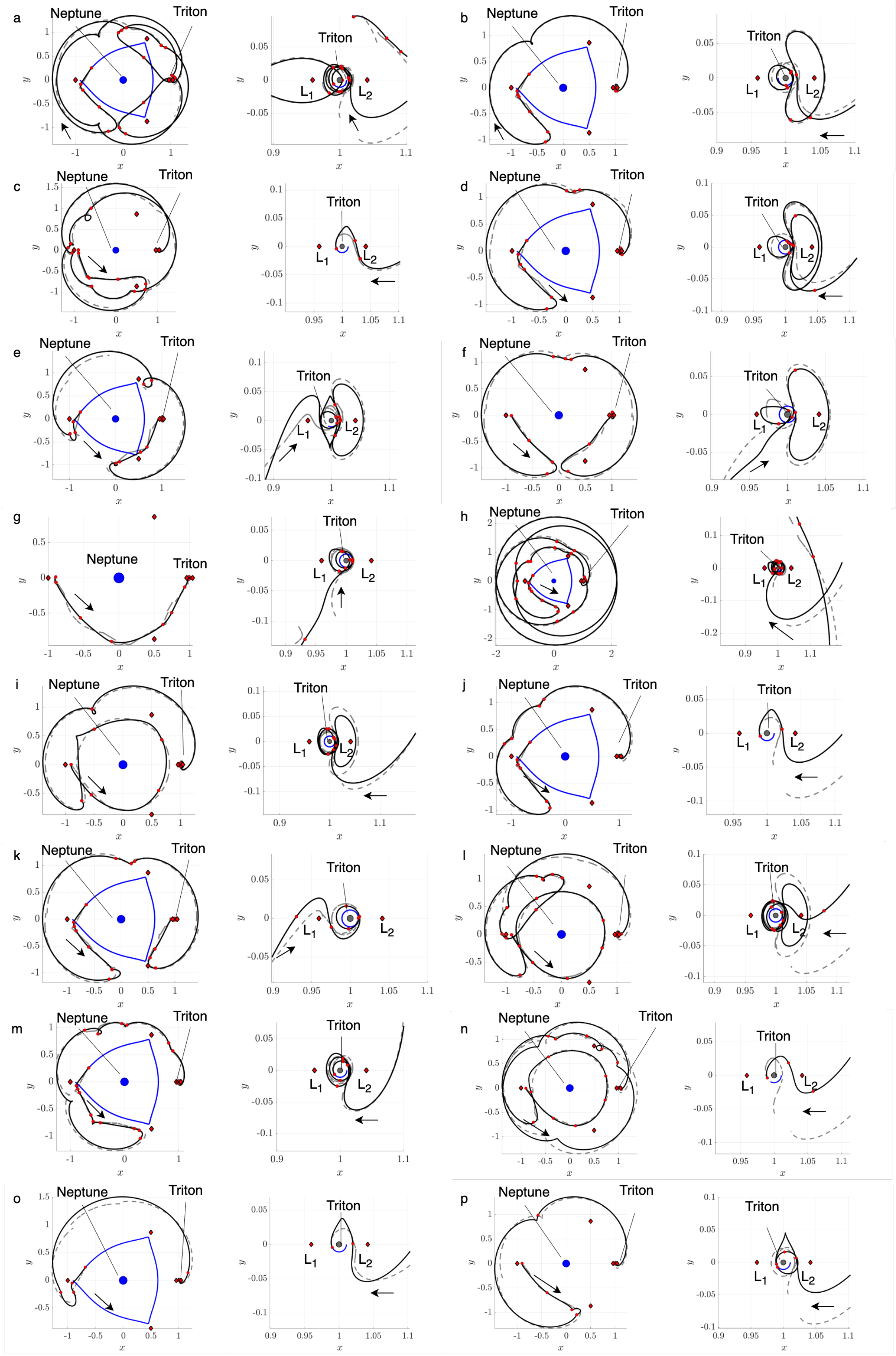


Figure 7.17: Final optimal trajectories for the low-energy trajectory design case

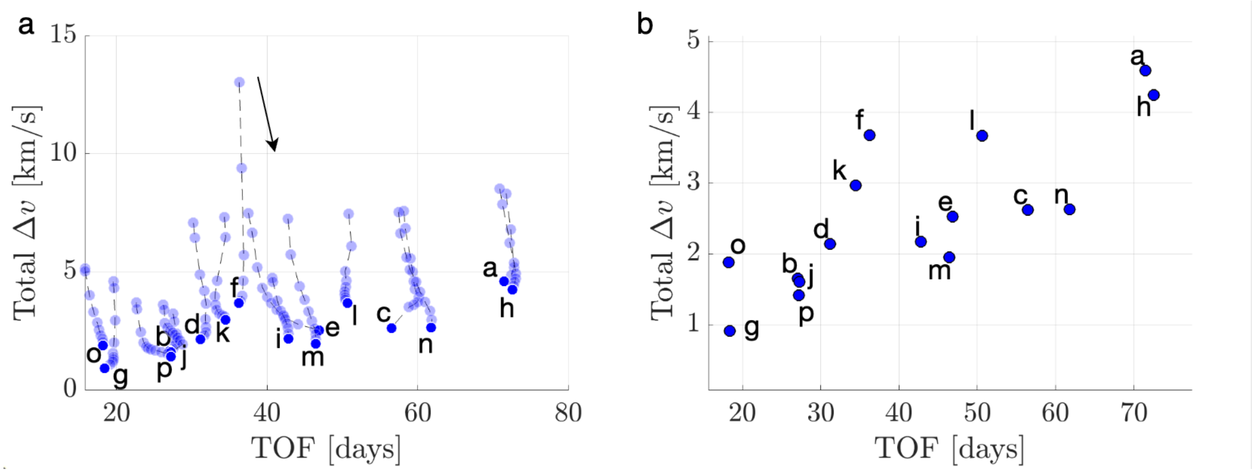


Figure 7.18: a) History of TOF and total Δv during the continuation steps to optimize each trajectory in Figure 7.17 b) Final TOF and total Δv for the trajectories in Figure 7.17

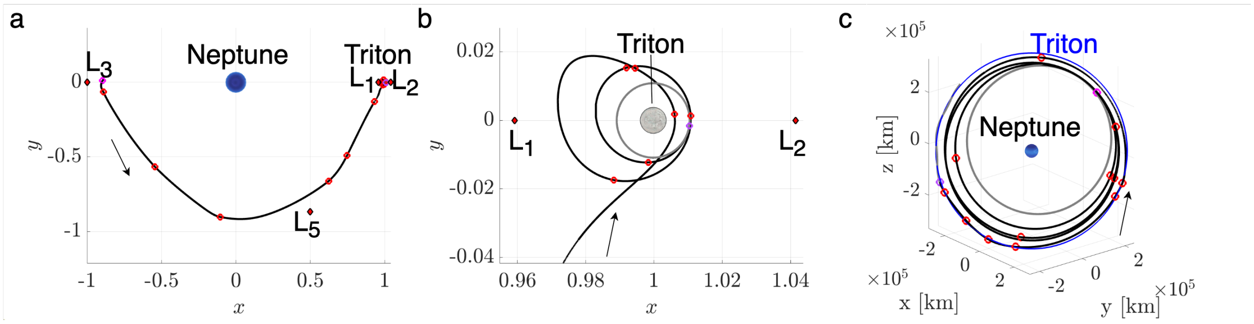


Figure 7.19: Trajectory in Figure 7.17g corrected in ephemeris model with a total $\Delta v = 0.9115$ km/s and TOF = 18.42 days, displayed (a) in the Neptune-Triton rotating frame with a global view of the trajectory around Neptune and (b) in the Neptune-Triton rotating frame with a zoomed-in view in the vicinity of Triton and c) in the ICRF centered on Neptune

Chapter 8

Concluding Remarks

8.1 Conclusions

This dissertation presents a data-driven approach to trajectory design for planetary missions. First, an approach to identify families of long-term stable orbits around the Moon is presented and applied to discover candidates for low lunar frozen orbits. Then, a motion primitive-based trajectory design approach is applied to the case of a mission to the Neptunian system, considered one of the future key targets of exploration in the 2023 NASA Decadal Survey [81].

In Chapters 4 and 5 of this dissertation, clustering is used to summarize a wide variety of lunar trajectories and identify candidates for low-lunar frozen and quasi-frozen orbits. First, a large set of trajectories is numerically propagated in a high-fidelity lunar gravity model with the point mass gravity of the Earth and Sun. Then, the evolution of the eccentricity and argument of perilune at a subset of perilunes along these trajectories is described using a finite-dimensional feature vector that captures its shape. These feature vectors are input to a hierarchical and density-based clustering algorithm, producing groups of trajectories with a similar geometry and phasing the evolution of perilune. From each cluster, the most tightly bounded evolution of perilune produces a trajectory that serves as the cluster representative. These representatives are then described by their boundary shape to create a geometric feature vector. The geometric feature vectors of the cluster representatives are grouped using a second clustering step. This grouping is used to merge clusters of trajectories with representatives that possess a similar geometry but distinct phasing. The result is a set of clusters of trajectories with a similar geometric evolution of perilune.

This dissertation applies the presented clustering-based framework to low lunar orbits generated from a wide variety of initial conditions with a semi-major axis of 1838 km at an initial epoch of January 1, 2025, 00:00.000 UTC. The resulting clusters of trajectories support the manual identification of 15 candidates for low lunar frozen orbits, each with a distinct geometry in their perilune evolution. Furthermore, the clusters support the identification of trends in the initial orbital elements that lead to changes in the orbit lifetime across a group of trajectories with a geometrically similar perilune evolution.

In the second part, this dissertation presents an extension of the motion primitive approach to trajectory design in multi-body systems, originally presented by Smith and Bosanac [101]. In robotics, motion primitives represent fundamental building blocks of motion that can be chained together to create a more complex trajectory. In their proof of concept, Smith and Bosanac extracted motion primitives from fundamental solutions of the Earth-Moon CR3BP using a consensus clustering approach. Then, they used the primitives to construct a graph, where each node represented a primitive and the edges connecting all the primitives were weighted considering the minimum discontinuity in position and velocity at nodes along the primitives. Then, using a breadth-first search approach, the graph was searched to obtain primitive sequences, which were refined with a sequential morphing and trimming approach to obtain trajectory initial guesses. Building on the previous approach, this work reformulates the steps of the motion primitive trajectory design framework to extend its applicability to more complex problems, such as planetary tours with path constraints.

First, motion primitives are obtained from periodic orbit and their stable and unstable invariant manifolds based on the geometric evolution of the trajectories. An analytical method, which selects consecutive orbits with the same number of curvature maxima, is leveraged to extract primitives from periodic orbits. Whereas a data-driven approach based on HDBSCAN and DBSCAN [41, 12] is used to obtain arcs with similar position and shape from the manifolds. Subsequently, the motion primitives are utilized to generate a graph that describes the flow across the selected set of primitives. In this new approach, the nodes of the graph are states sampled along the primitives.

The edge between two nodes is defined only if the two nodes meet the composability constraint, i.e., a discontinuity in position and velocity must be less than some values defining the nodes' proximity region. Constraints can be imposed in this step: constraints limiting the distance from a primary can remove members from the primitive $\mathcal{R}_e$, and therefore affect nodes of the graph, whereas constraints on velocity magnitude add another condition in the composability condition, i.e., the Δv between two nodes must be less than the imposed value.

The final graph is searched with a custom k-best search algorithm to produce diverse primitive sequences from the designated start and target nodes that minimize the variation of velocity direction across the nodes. The algorithm is based on a combination of Yen's algorithm [113] and A* [48] but adapted to retrieve solutions along primitive arcs. The sequences are the base for trajectory initial guesses. These are found by generating a graph that represents only the samples along the sequence's primitives and the members of their $\mathcal{R}_e$. Solutions are obtained as sequences of nodes that minimize the discontinuity in position. These initial guesses can be used in a correction and optimization scheme to obtain trajectories with impulsive maneuvers in the restricted problem and in an ephemeris model.

The updated motion primitive design approach is applied to three trajectory design scenarios in the Neptune-Triton system. The planetary system is modeled as a CR3BP to leverage the fundamental solutions governing the motion in the proximity of the primaries, Neptune and Triton. In the first case, a high-energy design is formulated to perform the Neptune Orbit Insertion from the incoming arc of an interplanetary transfer to a 1:7 resonant orbit, which could provide scientifically relevant observation of two of the smallest moons of Neptune, Proteus and Hyperion. The design results in a variety of initial guesses with various geometries and times of flight. These initial guesses are corrected and optimized, first in the NT-CR3BP and then in a point mass ephemeris model to obtain a solution trade space in total Δv and TOF.

In the second trajectory design case, the motion primitive approach is leveraged to retrieve trajectories within a medium range of Jacobi constant, from $C_J = 2.5$ to $C_J = 3$. The initial and final orbits, respectively a 4:3 resonant orbit and a 3:5 resonant orbit, provide two periapsis passages with Triton at a few thousand km of altitude per period, thus creating relevant scientific opportunities. As in the previous scenario, the motion primitive trajectory design approach is used to retrieve smooth and diverse initial guesses from selected primitives. These initial guesses are corrected and optimized to provide a solution space for the design problem.

Finally, the trajectory design process is repeated for a low-energy design case, using motion primitives characterized by a Jacobi constant $C_J > 3$. Here, the initial orbit is a 3:2 resonant orbit centered around Neptune, while the target of the transfer is a low prograde orbit around Triton. The solutions leverage the manifolds of the L_2 Lyapunov family to move from the Neptune-centered orbit to the LPO through the L_1 and L_2 gateways. Also in this case, the final trajectories are characterized by a variety of geometries, flight times, and maneuver costs, representing a large portion of the solution space.

8.2 Recommendations for Future Work

As the motion primitive approach to trajectory design is adapted to more complex scenarios in a multi-body system, a list of potential recommendations for future work includes:

- Incorporate the work from Gillespie, Miceli, and Bosanac [41] and Bosanac [12] to extract motion primitives from low-thrust and general trajectory arcs to expand the capability of the motion primitive design approach to generate initial guesses for more complex mission scenarios.

- The composability algorithm introduced in this research improved the connectivity of the graph, and consequently, the smoothness of the initial guesses. However, if a motion primitive and its $\mathcal{R}_e$ span a vast region of the proximity region around each sample of the primitive, it might become pretty significant. If the value of this proximity region does not reflect the actual composability of the node, the initial guess could be affected by substantial discontinuities. The computation of the nodes' composability region should be addressed such that the size of the primitive region does not influence the region's size.

- In the correction via collocation, a mesh is required to discretize the solution into a set of nodes, leveraged by the algorithm to retrieve a continuous trajectory. The mesh is generated from the primitive arcs by applying the same discretization over the entire initial guess. However, when the initial guess is composed of a series of primitive arcs of different sizes, the mesh might become too dense or too sparse, depending on the initial size of the arcs. To address this problem, which can slow down the correction step, an adaptive mesh size could be implemented. Nodes would be distributed along the motion primitive arc based on the length of the motion primitive arc. This improvement could significantly accelerate the correction of the motion primitive initial guesses.

- Explore the influence of maneuver placement on the trajectory's final geometry and cost in total Δv and time of flight.

- Incorporate other measures of trajectory characteristics that are desirable for planetary exploration and enable the fulfillment of additional science objectives.

- Apply the motion primitive trajectory design approach to other planetary systems and extend it to different dynamical models.

Bibliography

[1] Alberto Abad, Antonio Elipe, and Eva Tresaco. Analytical Model to Find Frozen Orbits for a Lunar Orbiter. Journal of Guidance, Control, and Dynamics, 32(3):888–898, 2009.

[2] Charles Acton. Ancillary Data Services of NASA's Navigation and Ancillary Information Facility. Planetary and Space Science, 44(1):65–70, 1996. Planetary data system.

[3] Charles Acton, Nathaniel Bachman, Boris Semenov, and Edward Wright. A Look Towards the Future in the Handling of Space Science Mission Geometry. Planetary and Space Science, 150:9–12, 2018. Enabling Open and Interoperable Access to Planetary Science and Heliophysics Databases and Tools.

[4] June Barrow-Green. Poincaré and the Three Body Problem, volume 11. American Mathematical Society, 1997.

[5] Mark Beckman. Mission Design for the Lunar Reconnaissance Orbiter. In 29th Annual AAS Guidance and Control Conference, Breckenridge, CO, February 2006.

[6] Richard Bellman. On a Routing Problem. Quarterly of Applied Mathematics, 16(1):87–90, 1958.

[7] Stefano Bonasera and Natasha Bosanac. Applying Data Mining Techniques to Higher-Dimensional Poincaré Maps in the Circular Restricted Three-Body Problem. Celestial Mechanics and Dynamical Astronomy, 133(11-12):51, December 2021.

[8] Natasha Bosanac. Leveraging Natural Dynamical Structures to Explore Multi-Body Systems. PhD Dissertation, Purdue University, West Lafayette, IN, 2016.

[9] Natasha Bosanac. Data-Mining Approach to Poincaré Maps in Multi-Body Trajectory Design. Journal of Guidance, Control, and Dynamics, 43(6):1190–1200, 2020.

[10] Natasha Bosanac. Data-Driven Summary of Natural Spacecraft Trajectories in the Earth-Moon System. In AAS/AIAA Astrodynamics Specialist Conference. Big Sky, MT, August 2023.

[11] Natasha Bosanac. Curvature Extrema Along Trajectories in the Circular Restricted Three-Body Problem. In AAS/AIAA Astrodynamics Specialists Conference, Broomfield, CO, 2024.

[12] Natasha Bosanac. Data-Driven Summary of Motion in an Ephemeris Model of Cislunar Space. In AAS/AIAA Space Flight Mechanics Meeting, Lihue, HI, 2025.

[13] Natasha Bosanac and Maxwell Joyner. Data-Driven Summary of Continuous Thrust Trajectories in a Low-Fidelity Model of Cislunar Space. In AAS/AIAA Astrodynamics Specialist Conference. Broomfield, CO, August 2024.

[14] William L Brogan. Modern Control Theory. Pearson Education India, 1985.

[15] Dirk Brouwer. Solution of the Problem of Artificial Satellite Theory Without Drag. Technical report, Yale University New Haven CT, United States, 1959.

[16] Stefano Campagnola, Brent B. Buffington, and Anastassios E. Petropoulos. Jovian Tour Design for Orbiter and Lander Missions to Europa. Acta Astronautica, 100:68–81, 2014.

[17] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In J. Pei, V.S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, Advances in Knowledge Discovery and Data Mining, page 160–172. Springer Berlin, Heidelberg, 2013.

[18] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 10(1):5, 2015.

[19] Bradley Cheetham, Thomas Gardner, Alec Forsman, Ethan Kayser, and Miekkal Clarkson. Capstone: A unique cubesat platform for a navigation demonstration in cislunar space. In ASCEND 2022, page 4382. 2022.

[20] Theodoros Chondrogiannis, Panagiotis Bouros, Johann Gamper, Ulf Leser, and David B. Blumenthal. Finding k-Shortest Paths with Limited Overlap. The International Journal on Very Large Data Bases, 29(5):1023–1047, 2020.

[21] Debora Clever, Monika Harant, Henning Koch, Katja Mombaur, and Dominik Endres. A Novel Approach for the Generation of Complex Humanoid Walking Sequences Based on a Combination of Optimal Control and Learning of Movement Primitives. Robotics and Autonomous Systems, 83:287–298, 2016.

[22] Shannon L Coffey, Andre Deprit, and Etienne Deprit. Frozen Orbits for Satellites Close to an Earth-Like Planet. Celestial Mechanics and Dynamical Astronomy, 59:37–72, 1994.

[23] Bruce A. Conway. Spacecraft Trajectory Optimization. Cambridge University Press, New York, 2010.

[24] Louis A. D'Amario, Larry E Bright, and Aron A. Wolf. Galileo Trajectory Design. Space Science Reviews, 60:23–78, 1992.

[25] Timothy F Dawn, Jeffrey Gutkowski, Amelia Batcha, Jacob Williams, and Samuel Pedrotty. Trajectory Design Considerations for Exploration Mission 1. In 2018 Space Flight Mechanics Meeting, page 0968, 2018.

[26] Carl De Boor. Good Approximation by Splines with Variable Knots II. In Proceedings of the Conference on the Numerical Solution of Differential Equations, Dundee, Scotland, 1973.

[27] Oriane Dermy, Alexandros Paraschos, Marco Ewerton, Jan Peters, François Charpillet, and Serena Ivaldi. Prediction of Intention during Interaction with iCub with Probabilistic Movement Primitives. Frontiers in Robotics and AI, 4, 2017.

[28] Edsger W Dijkstra. A Note on Two Problems in Connexion with Graphs. In Edsger Wybe Dijkstra: His Life, Work, and Legacy, pages 287–290. 2022.

[29] Antonio Elipe and Martin Lara. Frozen Orbits About the Moon. Journal of Guidance, Control, and Dynamics, 26(2):238–243, 2003.

[30] Todd A. Ely. Stable Constellations of Frozen Elliptical Inclined Lunar Orbits. The Journal of the Astronautical Sciences, 53:301–316, 2005.

[31] David Eppstein. Finding the k Shortest Paths. SIAM Journal on Computing, 28(2):652–673, 1998.

[32] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei 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, KDD'96, page 226–231. AAAI Press, 1996.

[33] Leigh N. Fletcher, Ravit Helled, Elias Roussos, Geraint Jones, Sébastien Charnoz, Nicolas André, David Andrews, Michele Bannister, Emma Bunce, Thibault Cavalié, Francesca Ferri, Jonathan Fortney, Davide Grassi, Léa Griton, Paul Hartogh, Ricardo Hueso, Yohai Kaspi, Laurent Lamy, Adam Masters, Henrik Melin, Julianne Moses, Oliver Mousis, Nadine Nettleman, Christina Plainaki, Jürgen Schmidt, Amy Simon, Gabriel Tobie, Paolo Tortora, Federico Tosi, and Diego Turrini. Ice Giant Systems: The Scientific Potential of Orbital Missions to Uranus and Neptune. Planetary and Space Science, 191:105030, 2020.

[34] William M Folkner, James G Williams, and Dale H Boggs. The Planetary and Lunar Ephemeris DE 421. Interplanetary Network Progress Report, 42-178:1–34, August 2009.

[35] David Folta, Natasha Bosanac, Ian Elliott, Laurie Mann, Rebecca Mesarch, and Jose Rosales. Astrodynamics Convention and Modeling Reference for Lunar, Cislunar, and Libration Point Orbits. Technical report, NASA, 2022.

[36] David Folta, Ken Galal, and David Lozier. Lunar Prospector Frozen Orbit Mission Design. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Boston, MA, August 1998.

[37] David Folta and David Quinn. Lunar Frozen Orbits. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, August 2006.

[38] Lester Randolph Ford and Delbert Ray Fulkerson. Flows in Networks. Princeton University Press, Princeton, NJ, 1962. ISBN: 9781400875184.

[39] Emilio Frazzoli. Robust Hybrid Control for Autonomous Vehicle Motion Planning. PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[40] Suman Gare, Soumita Chel, Priyanka D. Pantula, Abha Saxena, Kishalay Mitra, Rahuldeb Sarkar, and Lopamudra Giri. Analytics Pipeline for Visualization of Single Cell RNA Sequencing Data from Brochoaveolar Fluid in COVID-19 Patients: Assessment of Neuro Fuzzy-C-Means and HDBSCAN. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), pages 1634–1637, 2022.

[41] Cole Gillespie, Giuliana Elena Miceli, and Natasha Bosanac. Summarizing Natural and Controlled Motion in Cislunar Space With Behavioral Motion Primitives. In AAS/AIAA Space Flight Mechanics Meeting. Kauai, HI, January 2025.

[42] Daniel J. Grebow and Thomas A. Pavlak. MCOLL: Monte Collocation Trajectory Design Tool. AAS/AIAA Astrodynamics Specialist Conference, Stevenson, Washington, August 20-24, 2017, 2017.

[43] Matthew Greenhouse. The James Webb Space Telescope: Mission Overview and Status. In 2019 IEEE Aerospace Conference, pages 1–13, 2019.

[44] David J. Grymin, Charles B. Neas, and Mazen Farhood. A Hierarchical Approach for Primitive-Based Motion Planning and Control of Autonomous Vehicles. Robotics and Autonomous Systems, 62(2):214–228, 2014.

[45] Amanda F. Haapala and Kathleen C. Howell. Representations of Higher-Dimensional Poincaré Maps With Applications to Spacecraft Trajectory Design. Acta Astronautica, 96:23–41, March 2014.

[46] Alireza Hadjighasem, Daniel Karrasch, Hiroshi Teramoto, and George Haller. Spectral-Clustering Approach to Lagrangian Vortex Detection. Physical Review E, 93(6), 2016.

[47] Jiawei Han, Jian Pei, and Micheline Kamber. Data Mining: Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition edition, March 2006.

[48] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968.

[49] Michel Henon. Numerical Exploration of the Restricted Problem, V. Astronomy and Astrophysics, 1:223–238, 1969.

[50] George W. Hill. On the Part of the Motion of the Lunar Perigee which is a Function of the Mean Motions of the Sun and Moon. Acta Mathematica, 8:1–36, 1886.

[51] Anil N Hirani and Ryan P Russell. Approximations of distant retrograde orbits for mission design. In Spaceflight Mechanics 2006-AAS/AIAA Space Flight Mechnaics Meeting, pages 273–288, 2006.

[52] Kathleen C. Howell. Three-Dimensional Periodic Halo Orbits. Celestial Mechanics, 32(1):53, January 1984.

[53] John Jackson. Retrograde Satellite Orbits. Monthly Notices of the Royal Astronomical Society, Vol. 74, p. 62-82 (1913)., 74:62–82, 1913.

[54] Herbert B Keller. Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems. Applications of Bifurcation Theory, 1977.

[55] Lord Kelvin. LII. On Graphic Solution of Dynamical Problems. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 34(210):443–448, 1892.

[56] Wang Sang Koon, Martin W Lo, Jerrold E Marsden, and Shane D Ross. Dynamical Systems, The Three-Body Problem, and Space Mission Design. Springer, New York, USA, 2011.

[57] Yoshihide Kozai. Motion of a Lunar Orbiter. Publications of the Astronomical Society of Japan, 15(8):301, 1963.

[58] Dana Kulic, Wataru Takano, and Yoshihiko Nakamura. Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. The International Journal of Robotics Research, 27(7):761–784, 2008.

[59] Kenneth R Lang. Astrophysical Data: Planets and Stars. Springer Science & Business Media, 2012.

[60] Martin Lara. Design of Long-Lifetime Lunar Orbits: A Hybrid Approach. Acta Astronautica, 69(3):186–199, 2011.

[61] Martin Lara, Sebastián Ferrer, and Bernard De Saedeleer. Lunar Analytical Theory for Polar Orbits in a 50-Degree Zonal Model Plus Third-Body Effect. The Journal of the Astronautical Sciences, 57(3):561–577, 2009.

[62] Frank G. Lemoine, Sander Goossens, Terence J. Sabaka, Joseph B. Nicholas, Erwan Mazarico, David D. Rowlands, Bryant D. Loomis, Douglas S. Chinn, Gregory A. Neumann, David E. Smith, and Maria T. Zuber. GRGM900C: A Degree 900 Lunar Gravity Model From GRAIL Primary and Extended Mission Data. Geophysical Research Letters, 41(10):3382–3389, 2014.

[63] Alycia Leonard, Scot Wheeler, and Malcolm McCulloch. Power to the People: Applying Citizen Science and Computer Vision to Home Mapping for Rural Energy Access. International Journal of Applied Earth Observation and Geoinformation, 108:102748, 2022.

[64] Xiaodong Liu, Hexi Baoyin, and Xingrui Ma. Five Special Types of Orbits Around Mars. Journal of Guidance, Control, and Dynamics, 33(4):1294–1301, 2010.

[65] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. The International Journal of Robotics Research, 36(8):947–982, 2017.

[66] Mark Marley et al. Planetary Science Decadal Survey JPL Rapid Mission Architecture Neptune-Triton-KBO Study Final Report. Technical report, NASA Jet Propulsion Laboratory, Pasadena, CA, 2010.

[67] A. Masters, N. Achilleos, C.B. Agnor, S. Campagnola, S. Charnoz, B. Christophe, A.J. Coates, L.N. Fletcher, G.H. Jones, L. Lamy, F. Marzari, N. Nettelmann, J. Ruiz, R. Ambrosi, N. Andre, A. Bhardwaj, J.J. Fortney, C.J. Hansen, R. Helled, G. Moragas-Klostermeyer, G. Orton, L. Ray, S. Reynaud, N. Sergis, R. Srama, and M. Volwerk. Neptune and Triton: Essential pieces of the Solar System puzzle. Planetary and Space Science, 104:108–121, 2014.

[68] MathWorks. MATLAB. `https://www.mathworks.com/products/matlab.html`, 2025. Last accessed May 2025.

[69] Takehiko Matukuma. Periodic Orbits in Hill's Case. Third Paper (Periodic Ejectional Orbit). Proceedings of the Imperial Academy, 9(8):364–366, 1933.

[70] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical Density Based Clustering. Journal of Open Source Software, 2(11):205, 2017.

[71] Jeroen Melman, Giovanni Orlando, Ebrahim Safipour, Erwin Mooij, and Ron Noomen. Trajectory Optimization for a Mission to Neptune and Triton. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, page 7366, 2007.

[72] Giuliana E. Miceli, Natasha Bosanac, and Reza Karimi. Generating the Trajectory Design Space for Neptunian System Exploration. In AAS/AIAA Astrodynamics Specialist Conference. Broomfield, CO, August 2024.

[73] Giuliana E. Miceli, Natasha Bosanac, Michael A. Mesarch, David C. Folta, and Rebecca L. Mesarch. Clustering Approach To Identifying Low Lunar Frozen Orbits In A High-Fidelity Model. In AAS/AIAA Astrodynamics Specialist Conference. Big Sky, MT, August 2023.

[74] Giuliana Elena Miceli, Natasha Bosanac, Jeffrey R. Stuart, and Farah Alibay. Motion Primitive Approach to Spacecraft Trajectory Design in the Neptune-Triton System. In AIAA SciTech Forum. Orlando, FL, January 2024.

[75] Edward F Moore. The Shortest Path Through a Maze. In Proc. of the International Symposium on the Theory of Switching, pages 285–292. Harvard University Press, 1959.

[76] Davoud Moulavi, Pablo A. Jaskowiak, Ricardo J. G. B. Campello, Arthur Zimek, and Jörg Sander. Density-Based Clustering Validation. In Proceedings of the 2014 SIAM International Conference on Data Mining, pages 839–847. Society for Industrial and Applied Mathematics.

[77] O. Mousis, D.H. Atkinson, T. Cavalié, L.N. Fletcher, M.J. Amato, S. Aslam, F. Ferri, J.-B. Renard, T. Spilker, E. Venkatapathy, P. Wurz, K. Aplin, A. Coustenis, M. Deleuil, M. Dobrijevic, T. Fouchet, T. Guillot, P. Hartogh, T. Hewagama, M.D. Hofstadter, V. Hue, R. Hueso, J.-P. Lebreton, E. Lellouch, J. Moses, G.S. Orton, J.C. Pearl, A. Sánchez-Lavega, A. Simon, O. Venot, J.H. Waite, R.K. Achterberg, S. Atreya, F. Billebaud, M. Blanc, F. Borget, B. Brugger, S. Charnoz, T. Chiavassa, V. Cottini, L. d'Hendecourt, G. Danger, T. Encrenaz, N.J.P. Gorius, L. Jorda, B. Marty, R. Moreno, A. Morse, C. Nixon, K. Reh, T. Ronnet, F.-X. Schmider, S. Sheridan, C. Sotin, P. Vernazza, and G.L. Villanueva. Scientific Rationale for Uranus and Neptune in Situ Explorations. Planetary and Space Science, 155:12–40, 2018. Surfaces, atmospheres and magnetospheres of the outer planets, their satellites and ring systems: Part XII.

[78] Carl D Murray and Stanley F Dermott. Solar System Synamics. Cambridge University Press, 1999.

[79] Navid Nakhjiri and Benjamin Villac. Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design. Celestial Mechanics and Dynamical Astronomy, 123(1):63–83, 2015.

[80] NASA Navigation and Ancillary Information Facility (NAIF). NASA Navigation and Ancillary Information Facility (NAIF) Generic Kernels, 2023.

[81] National Academies of Sciences, Engineering, and Medicine. Nasa 2023 decadal survey. Technical report, National Academies Press, Washington, D.C., 2023.

[82] National Aeronautics and Space Administration (NASA). Neptune Fact Sheet, 2023.

[83] National Aeronautics and Space Administration (NASA). Neptunian Satellite Fact Sheet, 2023.

[84] Jorge Nocedal and Stephen J Wright. Numerical Optimization. Springer, 1999.

[85] Oystein Ore. Graphs and Their Uses, volume 34. Cambridge University Press, 1990.

[86] Hector Ortega-Arranz, Arturo Gonzalez-Escribano, and Diego R Llanos. The Shortest-Path Problem: Analysis and Comparison of Methods. Springer Nature, Switzerland, 2022.

[87] Aditya A Paranjape, Kevin C Meier, Xichen Shi, Soon-Jo Chung, and Seth Hutchinson. Motion Primitives and 3D Path Planning for Fast Flight Through a Forest. The International Journal of Robotics Research, 34(3):357–377, 2015. DOI: https://doi.org/10.1177/0278364914558017.

[88] Ryan S. Park, William M. Folkner, James G. Williams, and Dale H. Boggs. The JPL Planetary and Lunar Ephemerides DE440 and DE441. The Astronomical Journal, 161(3):105, feb 2021.

[89] Sang-Young Park and John Junkins. Orbital Mission Analysis for a Lunar Mapping Satellite. In Astrodynamics Conference, page 3717, 1994.

[90] Marci E. Paskowitz and Daniel J. Scheeres. Robust Capture and Transfer Trajectories for Planetary Satellite Orbiters. Journal of Guidance, Control, and Dynamics, 29(2):342–353, 2006.

[91] Nicholas M Patrikalakis and Takashi Maekawa. Shape Interrogation for Computer Aided Design and Manufacturing, volume 15. Springer, 2002.

[92] Robert E. Pritchett. Strategies for Low-Thrust Transfer Design Based on Direct Collocation Techniques. Thesis, Purdue University Graduate School, 2020.

[93] Lars Reng, Thomas B Moeslund, and Erik Granum. Finding Motion Primitives in Human Body Gestures. In Gesture in Human-Computer Interaction and Simulation: 6th International Gesture Workshop, pages 133–144, Berder Island, France, 05 2005. DOI: https://doi.org/10.1007/11678816_16.

[94] Malin Rosengren. Improved Technique for Passive Eccentricity Control. Orbital Mechanics and Mission Design, pages 49–58, January 1989.

[95] R.D. Russell and J. Christiansen. Adaptive Mesh Selection Strategies for Solving Boundary Value Problems. SIAM Journal on Numerical Analysis, 15(1):59–80, 1978.

[96] Ryan P. Russell and Martin Lara. Long-Lifetime Lunar Repeat Ground Track Orbits. Journal of Guidance, Control, and Dynamics, 30(4):982–993, 2007.

[97] Abigail M. Rymer, Kirby D. Runyon, Brenda Clyde, Jorge I. Núñez, Romina Nikoukar, Krista M. Soderlund, Kunio Sayanagi, Mark Hofstadter, Lynnae C. Quick, S. Alan Stern, Tracy Becker, Matthew Hedman, Ian Cohen, Frank Crary, Jonathan J. Fortney, Janet Vertesi, Candy Hansen, Imke de Pater, Carol Paty, Thomas Spilker, Tom Stallard, George B. Hospodarsky, H. Todd Smith, Hannah Wakeford, Sarah E. Moran, Andrew Annex, Paul Schenk,

Martin Ozimek, Juan Arrieta, Ralph L. McNutt, Adam Masters, Amy A. Simon, Susan Ensor, Clint T. Apland, Jonathan Bruzzi, D. Alex Patthoff, Christopher Scott, Christian Campo, Christopher Krupiarz, Corey J. Cochrane, Curt Gantz, Dan Rodriguez, Dan Gallagher, Dana Hurley, Doug Crowley, Elizabeth Abel, Elena Provornikova, Elizabeth P. Turtle, George Clark, Jacob Wilkes, Jack Hunt, James H. Roberts, Jeremy Rehm, Kelvin Murray, Larry Wolfarth, Leigh N. Fletcher, Linda Spilker, Emily S. Martin, Marzia Parisi, Mike Norkus, Noam Izenberg, Robert Stough, Ron J. Vervack, Kathleen Mandt, Kevin B. Stevenson, Seth Kijewski, Weilun Cheng, Jay D. Feldman, Gary Allen, Dinesh Prabhu, Soumya Dutta, Cindy Young, and Joseph Williams. Neptune Odyssey: A Flagship Concept for the Exploration of the Neptune–Triton System. The Planetary Science Journal, 2(5):184, 2021.

[98] Felix San-Juan, Alberto Abad, Antonio Elipe, and Eva Tresaco. Analytical Model for Lunar Orbiter. Advances in the Astronautical Sciences, 130:1669–1680, January 2008.

[99] Daniel Jay Scheeres. Orbit Mechanics About Asteroids and Comets. Journal of Guidance, Control, and Dynamics, 35(3):987–997, 2012.

[100] Thomas R. Smith and Natasha Bosanac. Constructing Motion Primitive Sets to Summarize Periodic Orbit Families and Hyperbolic Invariant Manifolds in a Multi-Body System. Celestial Mechanics and Dynamical Astronomy, 134(1):7, 2022.

[101] Thomas R. Smith and Natasha Bosanac. Motion Primitive Approach to Spacecraft Trajectory Design in a Multi-body System. The Journal of the Astronautical Sciences, 70(34), 2023.

[102] Byron Swenson. Neptune Atmospheric Probe Mission. In Guidance, Navigation and Control Conference. American Institute of Aeronautics and Astronautics, 1992.

[103] Victor Szebehely. Theory of Orbits: The Restricted Problem of Three Bodies. Academic Press, London, 1967.

[104] Diego Turrini, Romolo Politi, Roberto Peron, Davide Grassi, Christina Plainaki, Mauro Barbieri, David M. Lucchesi, Gianfranco Magni, Francesca Altieri, Valeria Cottini, Nicolas Gorius, Patrick Gaulme, François-Xavier Schmider, Alberto Adriani, and Giuseppe Piccioni. The Comparative Exploration of the Ice Giant Planets with Twin Spacecraft: Unveiling the History of Our Solar System. Planetary and Space Science, 104:93–107, 2014.

[105] Stella Tzirti, Kleomenis Tsiganis, and Harry Varvoglis. Effect of 3rd-Degree Gravity Harmonics and Earth Perturbations on Lunar Artificial Satellite Orbits. Celestial Mechanics and Dynamical Astronomy, 108(4):389–404, 2010.

[106] David A. Vallado. Fundamentals of Astrodynamics and Applications, Fifth Edition. Microcosm Press, New York, 2022.

[107] Tatiana Mar Vaquero Escribano. Spacecraft Transfer Trajectory Design Exploiting Resonant Orbits in Multi-Body Environments. PhD thesis, Purdue University, West Lafayette, IN, 2013.

[108] Benjamin F Villac, Rodney L Anderson, and Alex J Pini. Computer Aided Ballistic Orbit Classification Around Small Bodies. Journal of Astronautical Sciences, 63(3):175–205, 2016.

[109] Boyang Wang, Jianwei Gong, Ruizeng Zhang, and Huiyan Chen. Learning to Segment and Represent Motion Primitives from Driving Data for Motion Planning Applications. In 21st International Conference on Intelligent Transportation Systems, pages 1408–1414, Maui, HI, November 2018. DOI: https://doi.org/10.1109/ITSC.2018.8569913.

[110] Takeshi Watanabe, Tomoaki Tatsukawa, Takayuki Yamamoto, Akira Oyama, and Yasuhiro Kawakatsu. Design Exploration of Low-Thrust Space Trajectory Problem for DESTINY Mission. Journal of Spacecraft and Rockets, 54(4):796–807, 2017.

[111] Paul Williams. Hermite-Legendre-Gauss-Lobatto Direct Transcription in Trajectory Optimization. Journal of Guidance, Control, and Dynamics, 32(4):1392–1395, 2009.

[112] Artur Wolek and Craig A. Woolsey. Model-Based Path Planning, pages 183–206. Springer International Publishing, Cham, 2017.

[113] Jin Yen. An Algorithm for Finding Shortest Routes from All Source Nodes to a Given Destination in General Networks. Quarterly of Applied Mathematics, 27:526–530, 1970.

Appendix A

Motion Primitives for Periodic Orbits and Hyperbolic Invariant Manifolds in the Neptune-Triton CR3BP

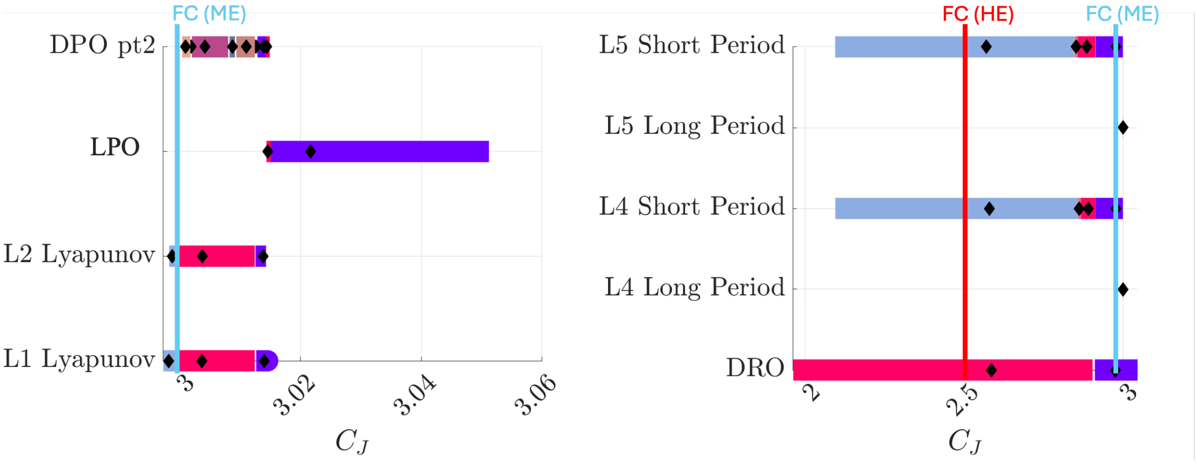


Figure A.1: Jacobi constant range for P_2 centered motion primitives and their region of existence. The primitives are indicated with a black diamond marker inside their respective region of existence bar. The region of existence colors correspond to the primitive colors in Figure A.6. Lines indicate the initial (IC) and final (FC) conditions for the high-energy (HE) transfer in red, and medium-energy (ME) transfer in blue.

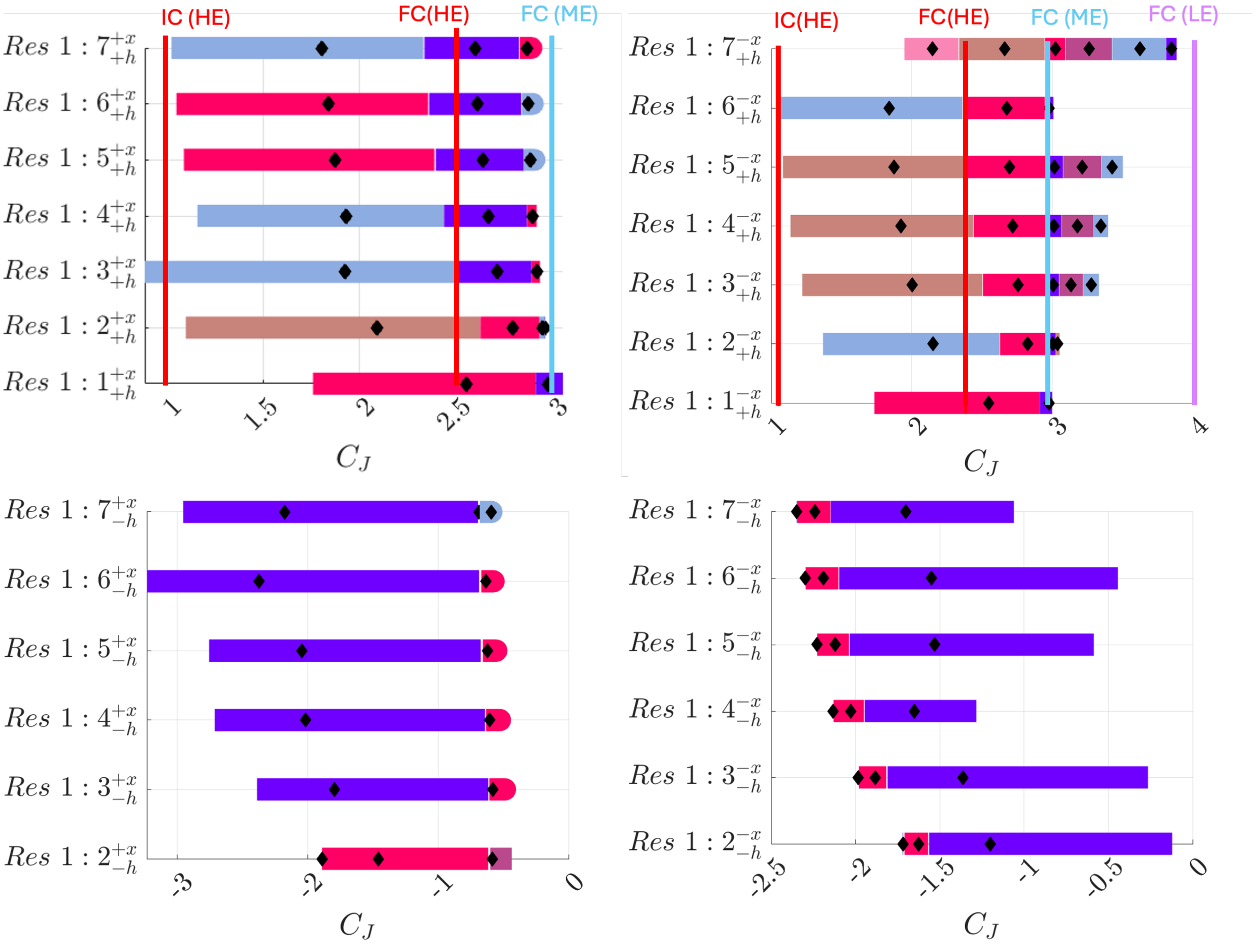


Figure A.2: Jacobi constant range for 1:q resonant motion primitives and their region of existence, where $q = [1, 2, 3, 4, 5, 6, 7]$. The primitives are indicated with a black diamond marker inside their respective region of existence bar. The region of existence colors correspond to the primitive colors in Figures A.7 and A.8. Lines indicate the initial (IC) and final (FC) conditions for the high-energy (HE) transfer in red, medium-energy (ME) transfer in blue, and low-energy (LE) transfer in pink.

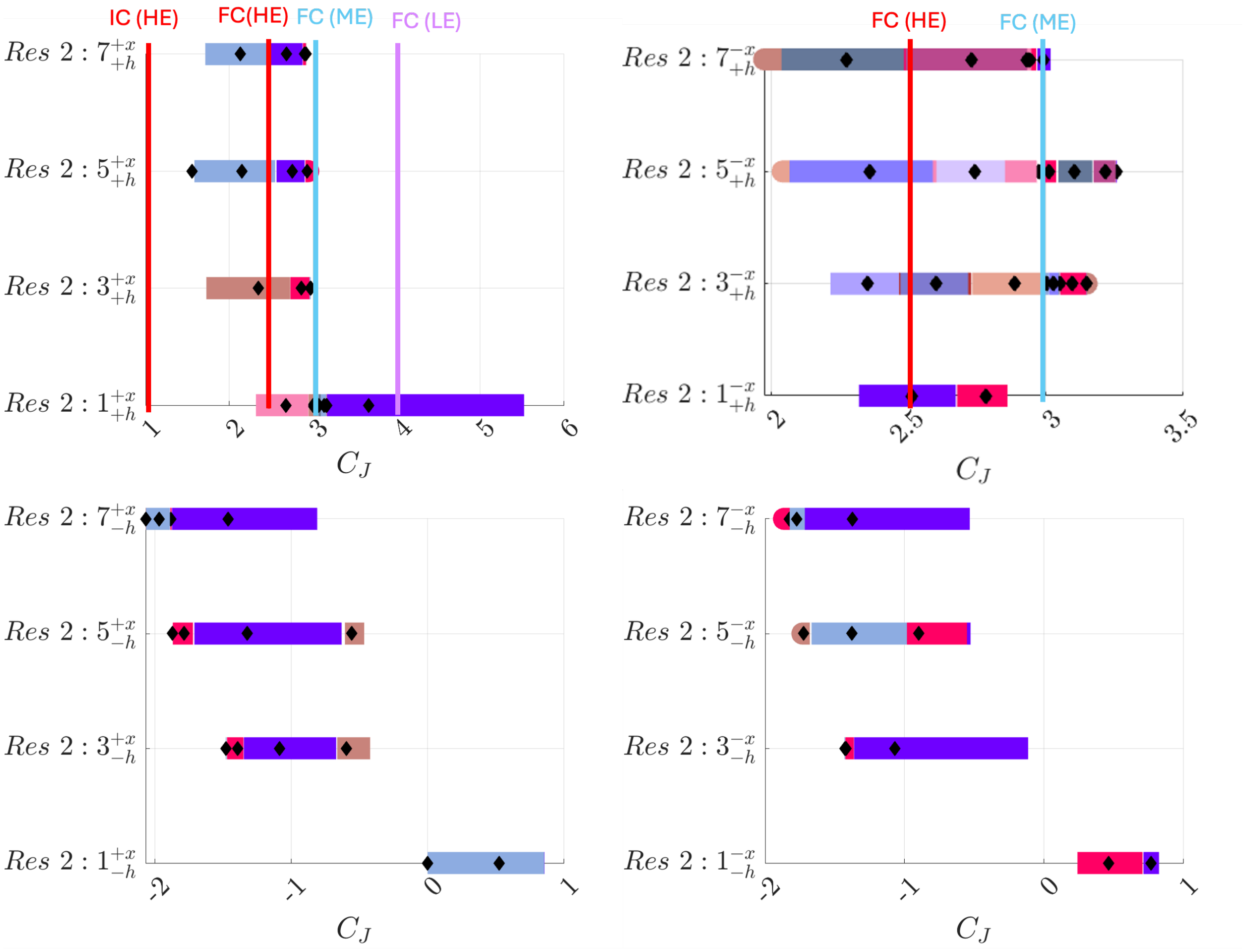


Figure A.3: Jacobi constant range for 2:q resonant motion primitives and their region of existence, where $q = [1, 3, 5, 7]$. The primitives are indicated with a black diamond marker inside their respective region of existence bar. The region of existence colors correspond to the primitive colors in Figure A.9. Lines indicate the initial (IC) and final (FC) conditions for the high-energy (HE) transfer in red, medium-energy (ME) transfer in blue, and low-energy (LE) transfer in pink.

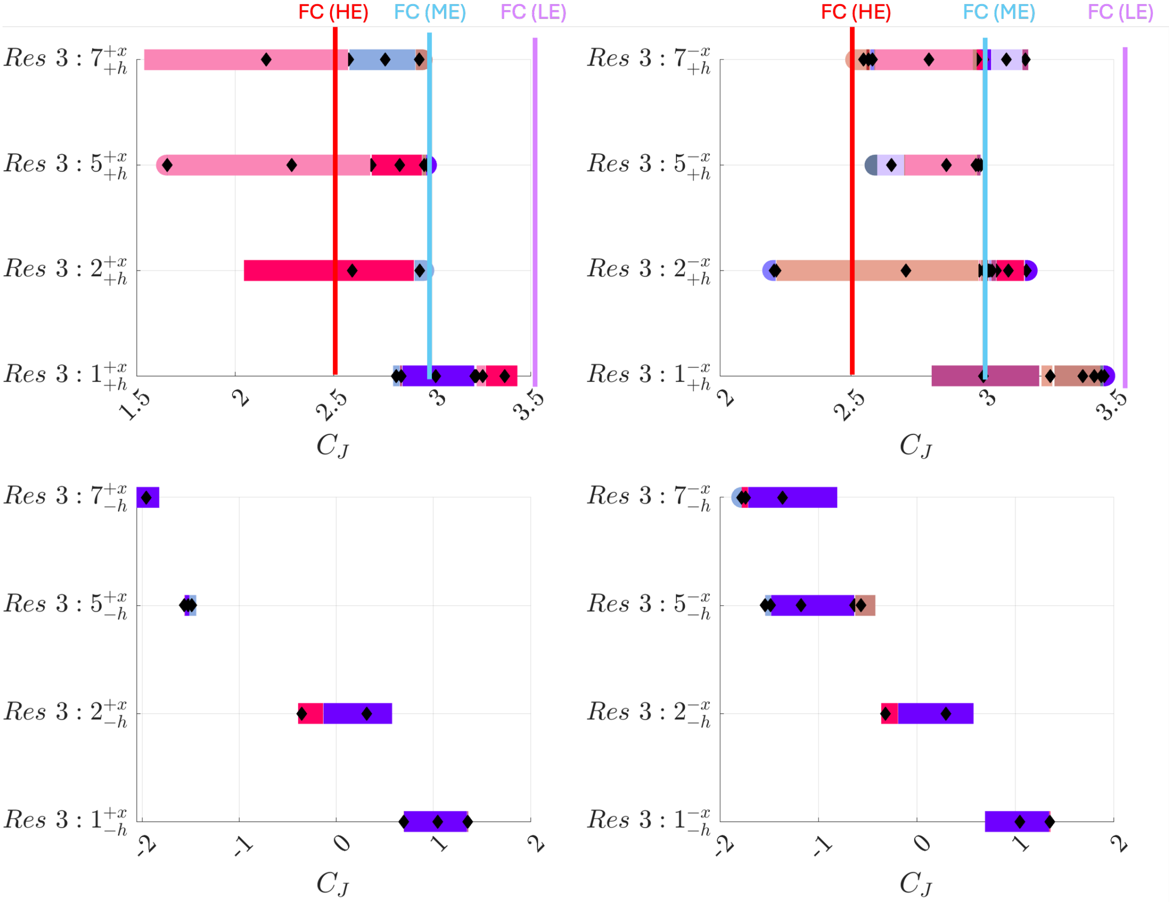


Figure A.4: Jacobi constant range for 3:q resonant motion primitives and their region of existence, where $q = [1, 2, 5, 7]$. The primitives are indicated with a black diamond marker inside their respective region of existence bar. The region of existence colors correspond to the primitive colors in Figure A.10. Lines indicate the initial (IC) and final (FC) conditions for the high-energy (HE) transfer in red, medium-energy (ME) transfer in blue, and low-energy (LE) transfer in pink.

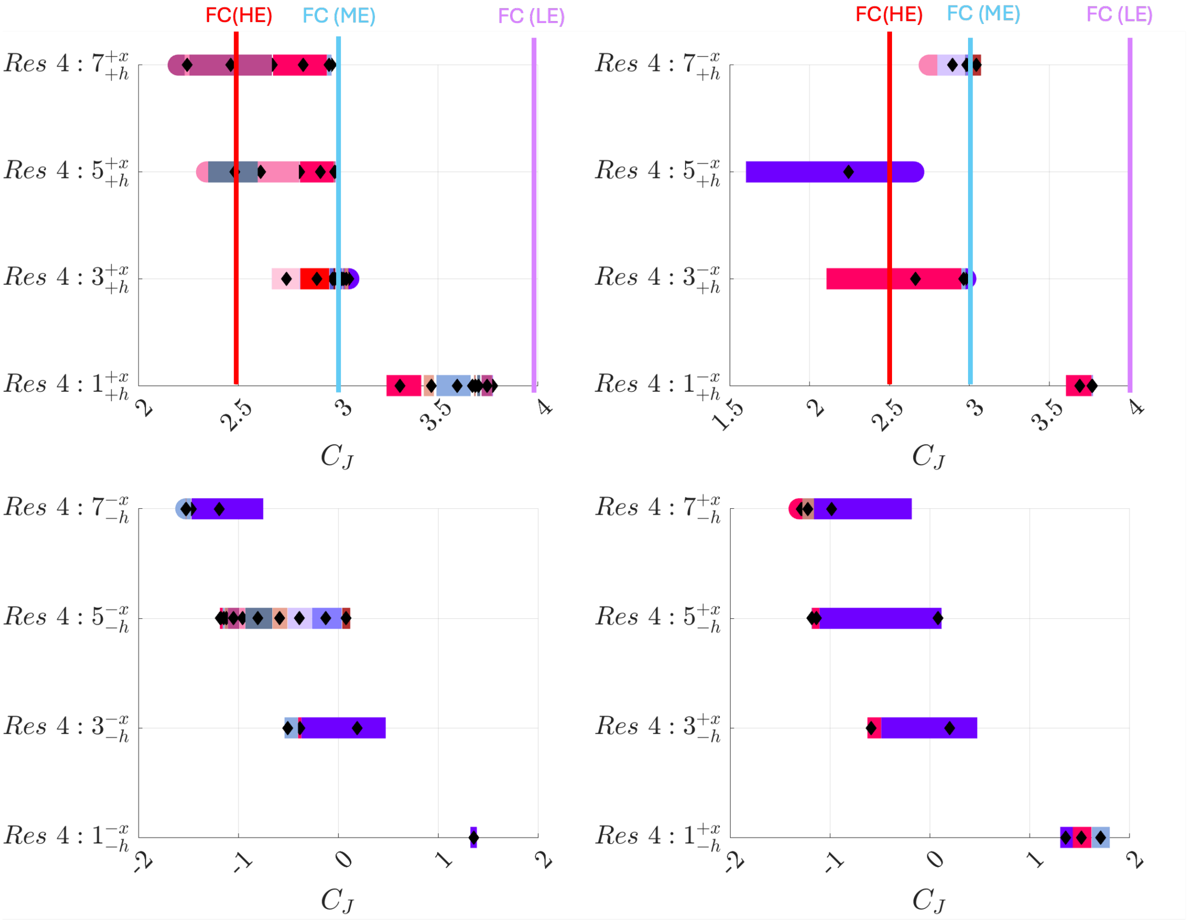


Figure A.5: Jacobi constant range for 4:q resonant motion primitives and their region of existence, where $q = [1, 3, 5, 7]$. The primitives are indicated with a black diamond marker inside their respective region of existence bar. The region of existence colors correspond to the primitive colors in Figure A.11. Lines indicate the initial (IC) and final (FC) conditions for the high-energy (HE) transfer in red, medium-energy (ME) transfer in blue, and low-energy (LE) transfer in pink.

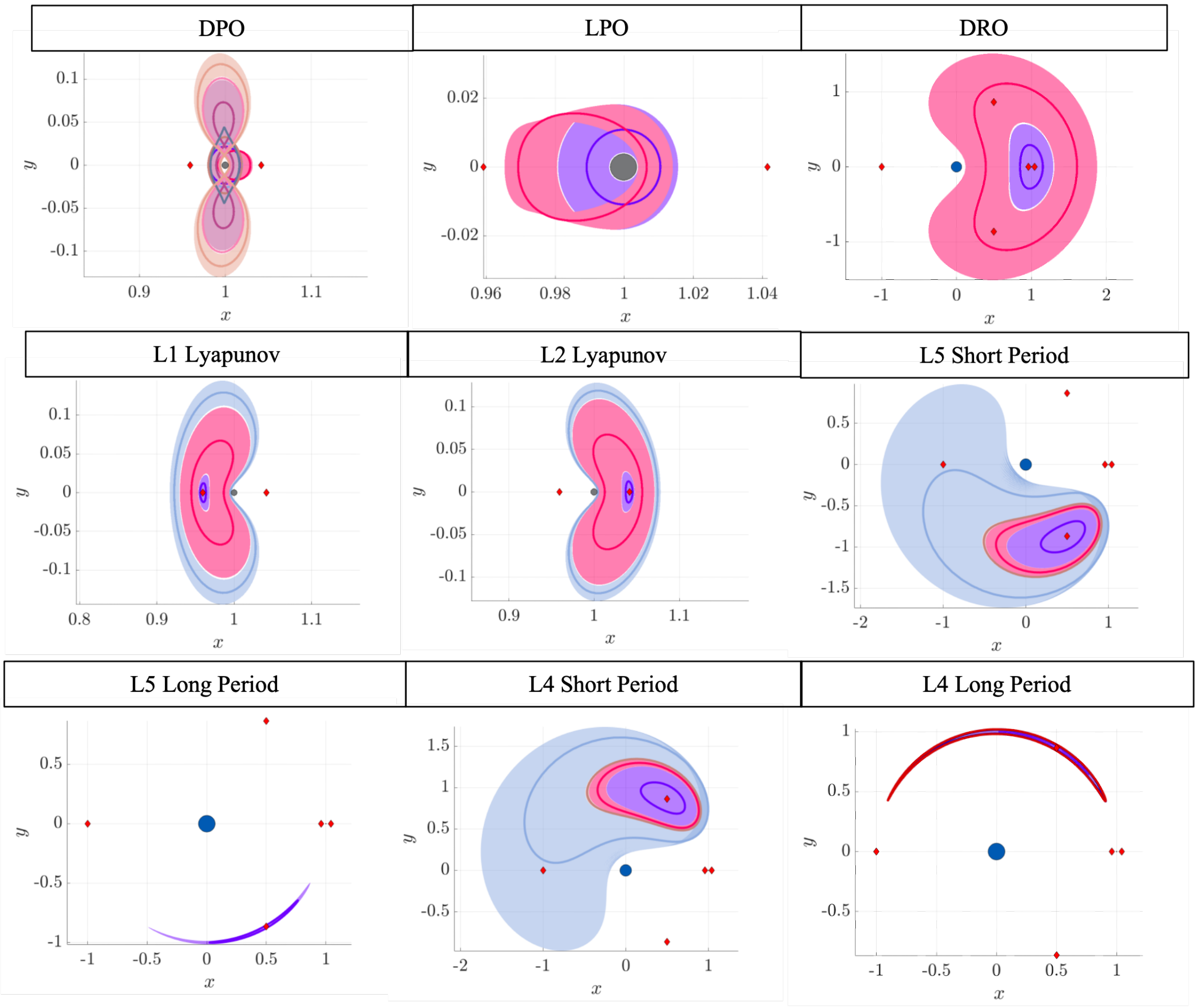


Figure A.6: Motion primitives obtained for P_2 centered periodic orbit families. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.

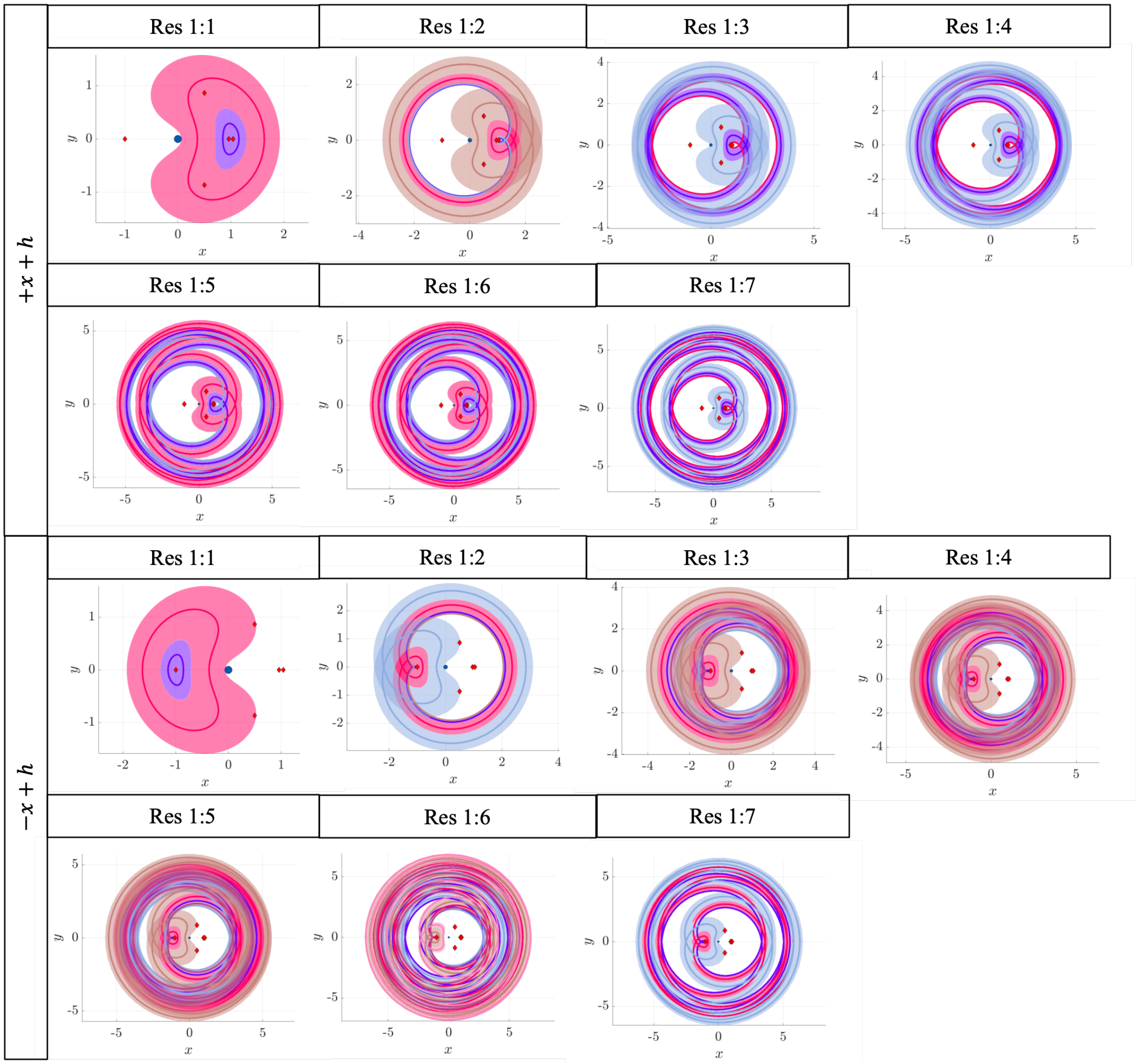
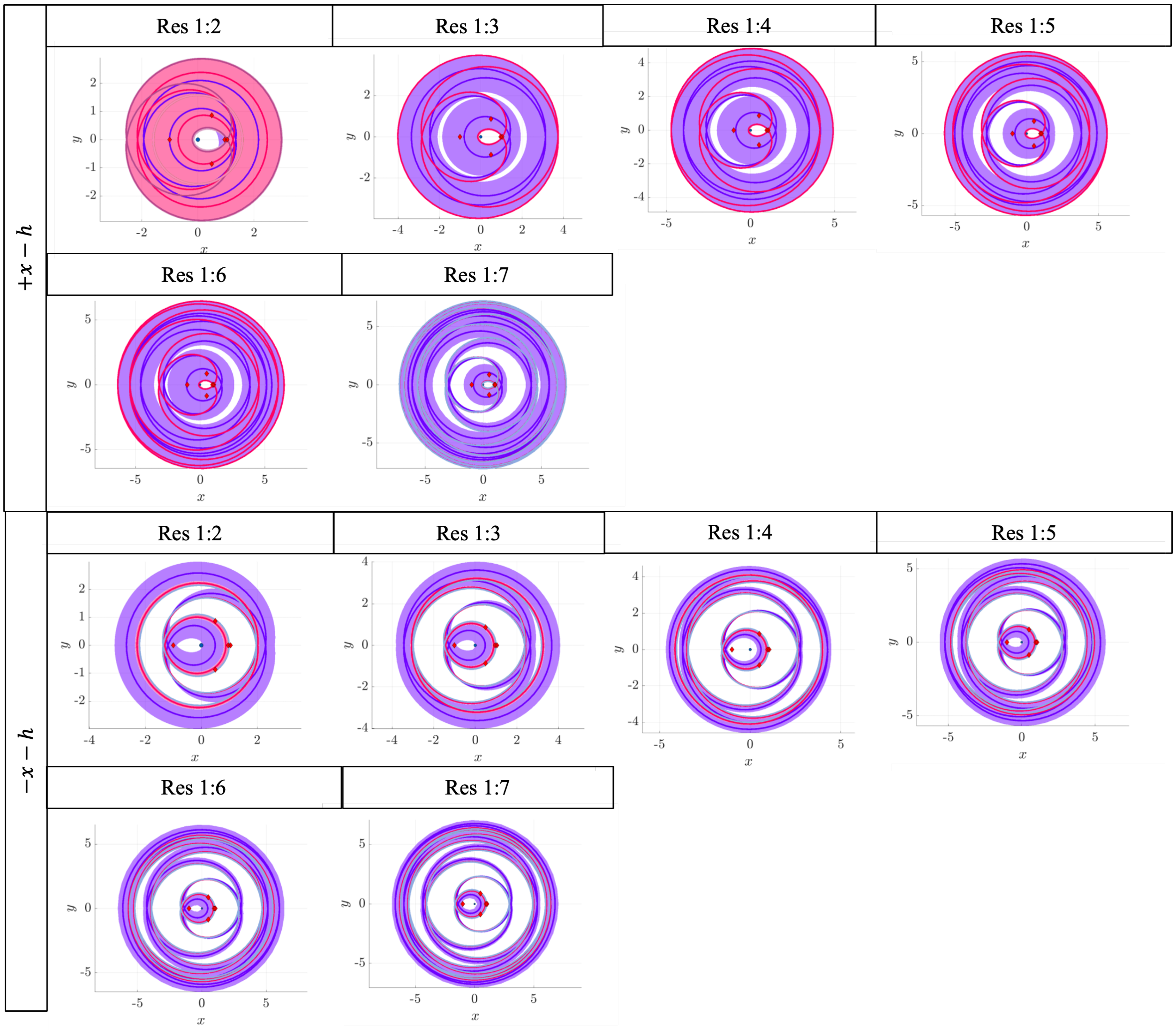


Figure A.7: Motion primitives obtained for prograde 1:q resonant orbit families, where $q = [1, 2, 3, 4, 5, 6, 7]$. The labels on the left indicate the location and direction of motion of the periapsis state used to generate the family, as detailed in Section 2.2.2.2. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.


+x − h

Res 1:2

Res 1:3

Res 1:4

Res 1:5

Res 1:6

Res 1:7

−x − h

Res 1:2

Res 1:3

Res 1:4

Res 1:5

Res 1:6

Res 1:7

Figure A.8: Motion primitives obtained for retrograde 1:q resonant orbit families, where $q = [1, 2, 3, 4, 5, 6, 7]$. The labels on the left indicate the location and direction of motion of the periapsis state used to generate the family, as detailed in Section 2.2.2.2. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.

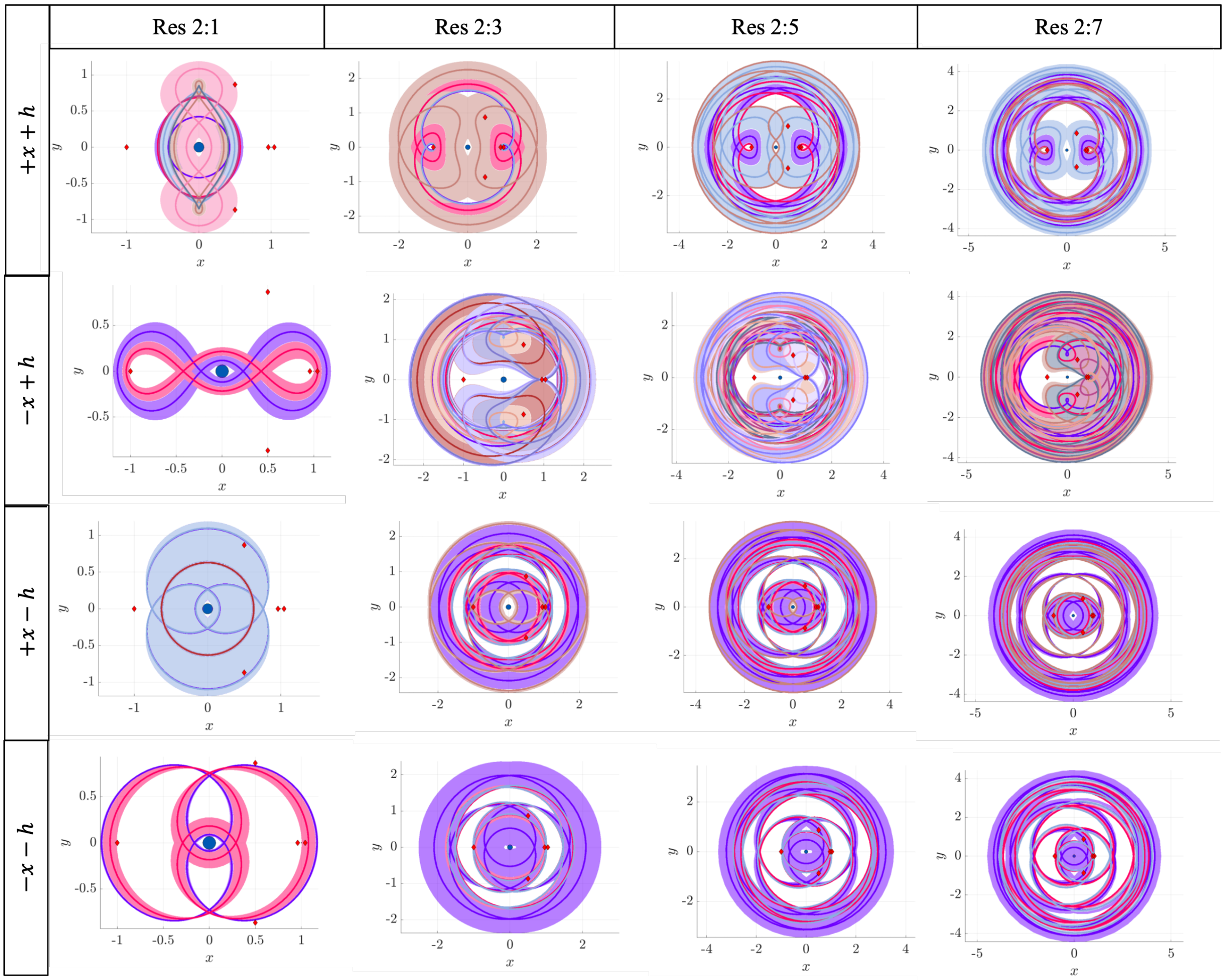


Figure A.9: Motion primitives obtained for 2:q resonant orbit families, where $q = [1, 3, 5, 7]$. The labels on the left indicate the location and direction of motion of the periapsis state used to generate the family, as detailed in Section 2.2.2.2. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.

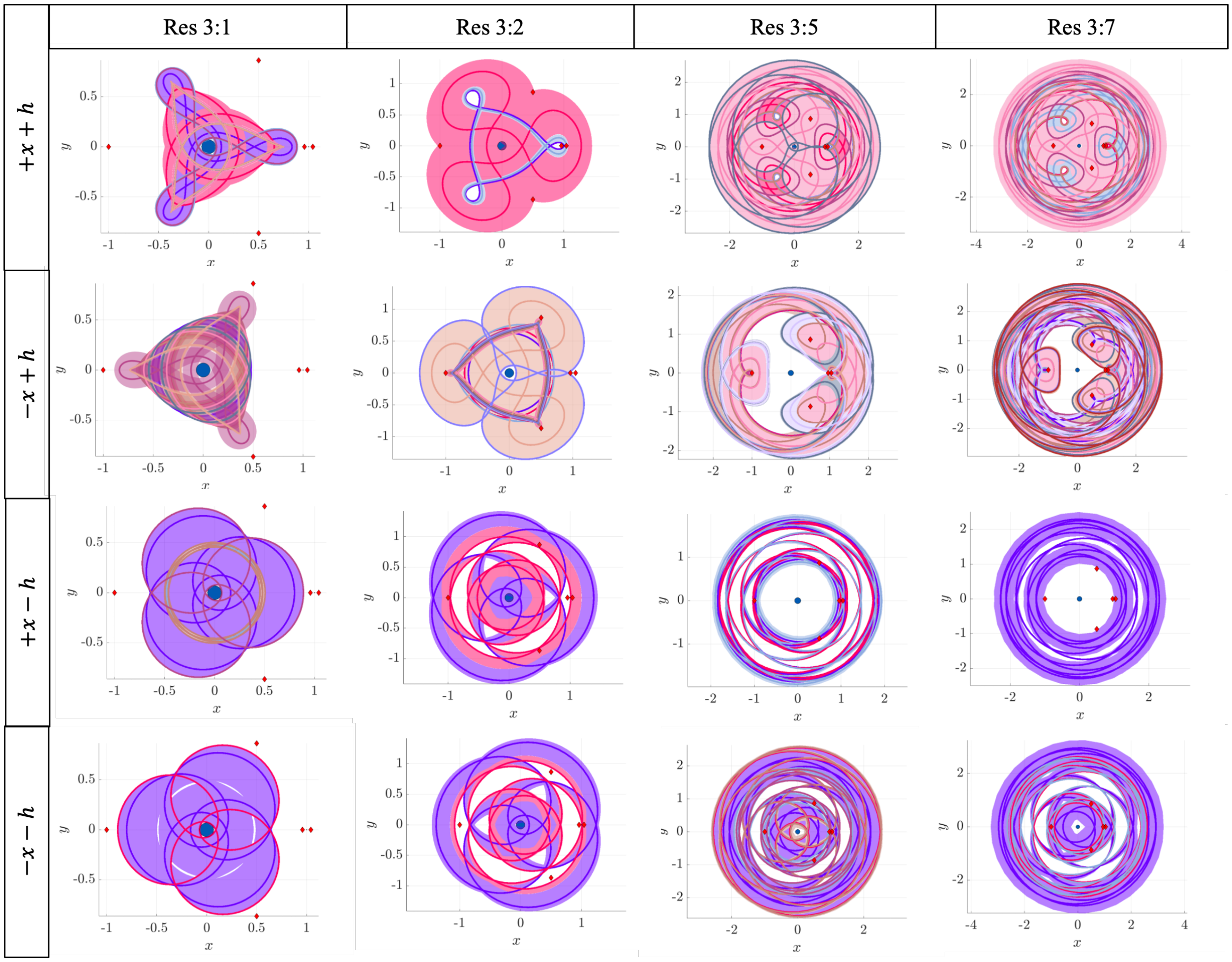


Figure A.10: Motion primitives obtained for 3:q resonant orbit families, where $q = [1, 2, 5, 7]$. The labels on the left indicate the location and direction of motion of the periapsis state used to generate the family, as detailed in Section 2.2.2.2. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.

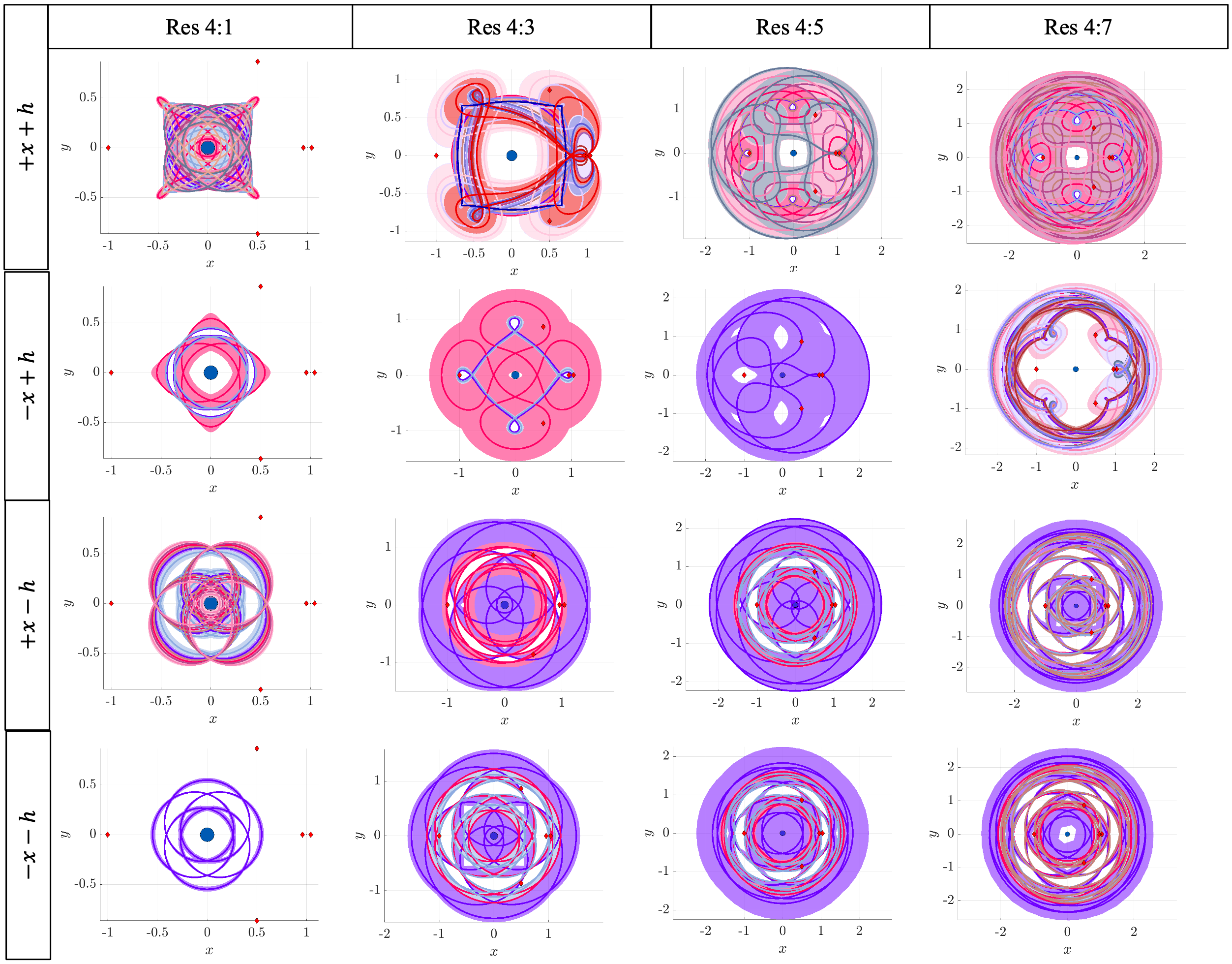


Figure A.11: Motion primitives obtained for 4:q resonant orbit families, where $q = [1, 3, 5, 7]$. The labels on the left indicate the location and direction of motion of the periapsis state used to generate the family, as detailed in Section 2.2.2.2. Each primitive set is highlighted with a unique color. In each set, the thick line represents the motion primitive, while the region of existence is represented as a shaded region of the same color of the primitive.

Appendix B

Lunar Frozen Orbits

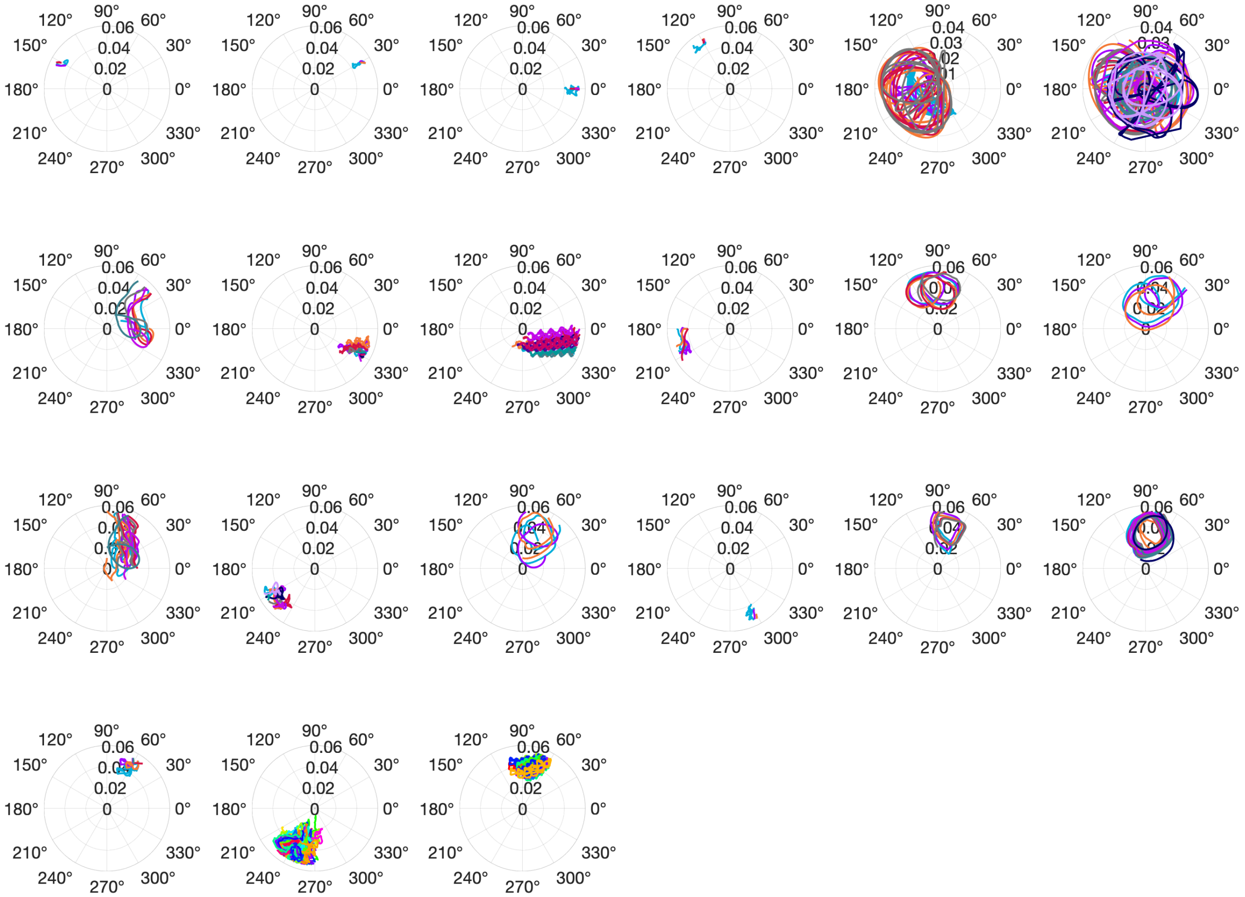


Figure B.1: Perilune evolution in the $e-\omega$ polar plot for representative members of clusters in $\mathcal{C}_g$. Most of these groups do not contain candidates for frozen and quasi-frozen orbits, whereas others contain candidates for long-term bounded orbits.

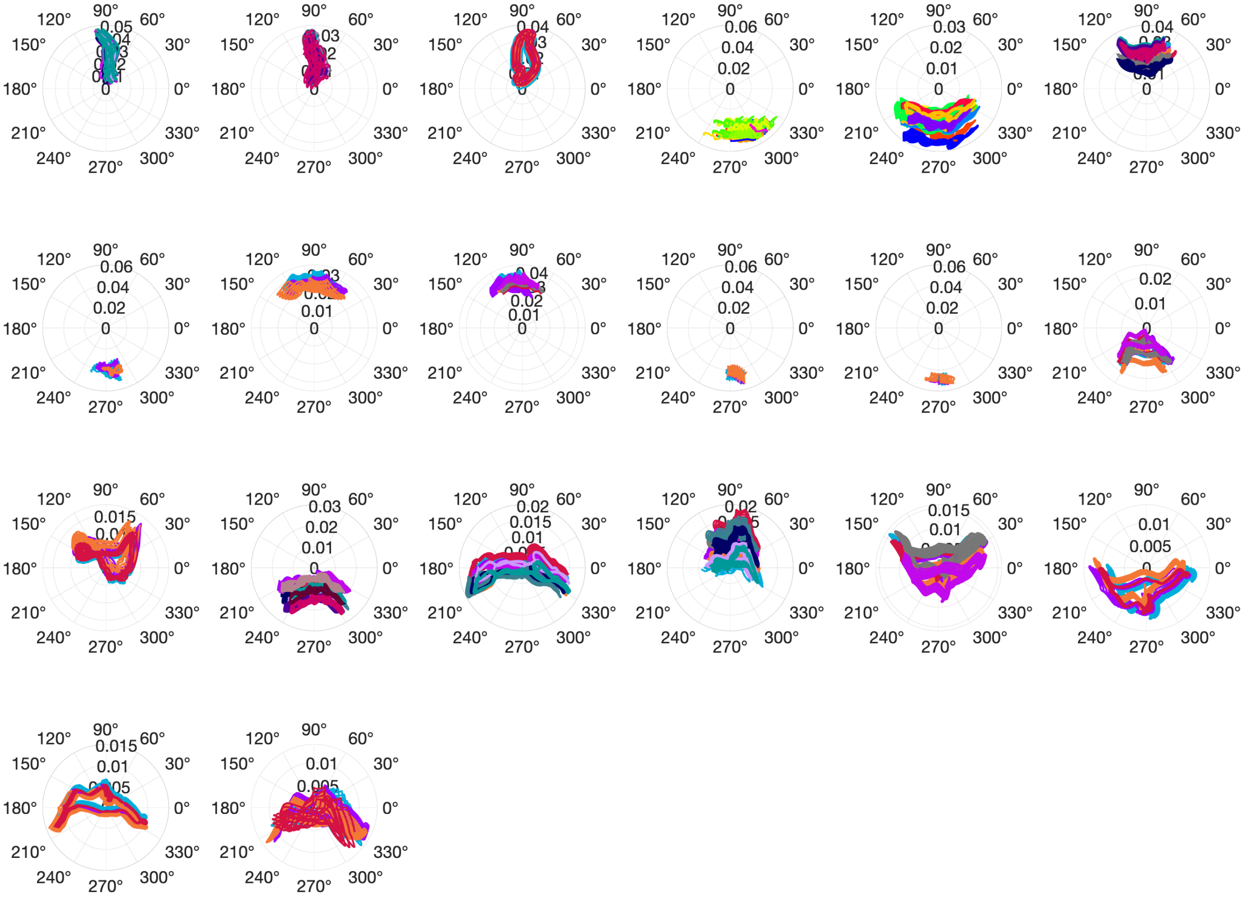


Figure B.2: Perilune evolution in the $e - \omega$ polar plot for representative members of clusters in $\mathcal{C}_g$ containing candidates for frozen and quasi frozen orbits.

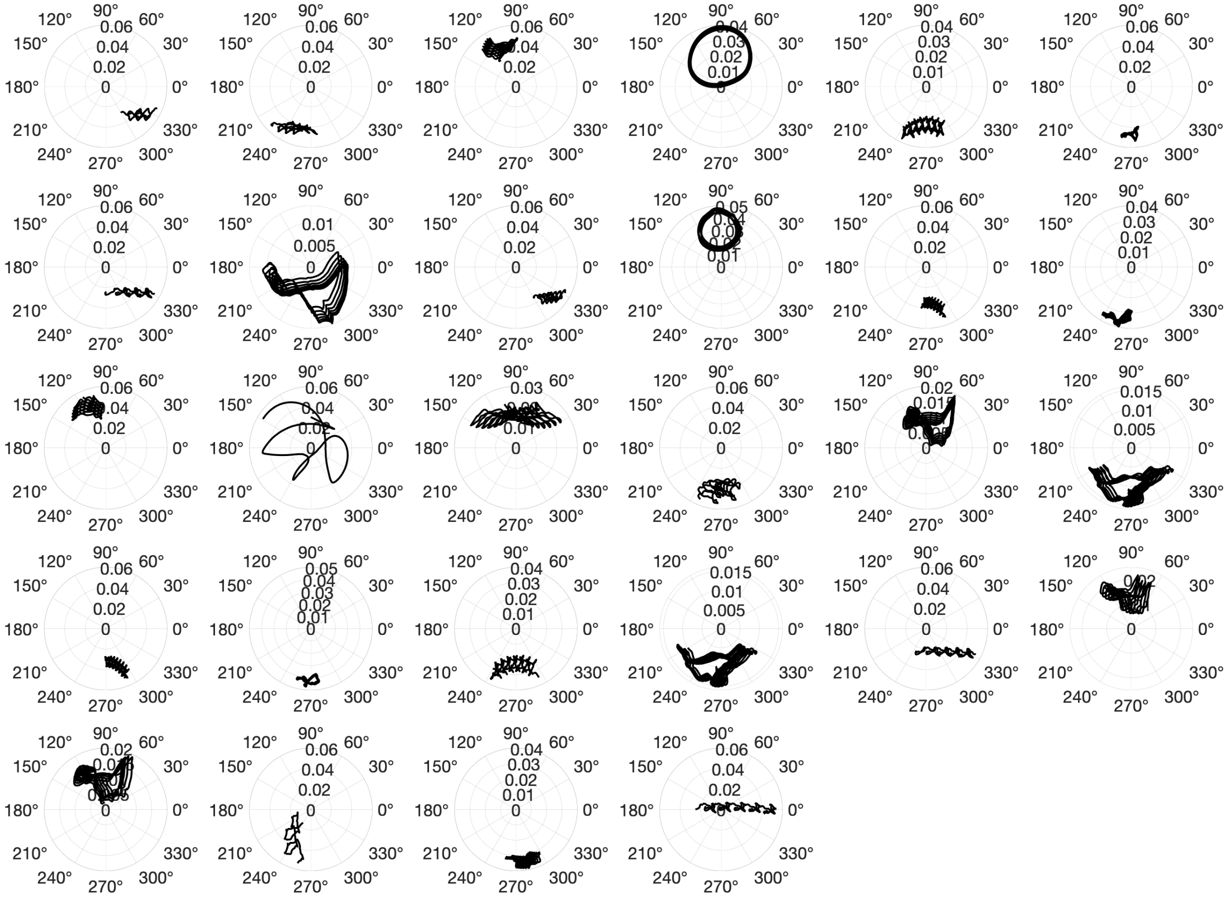


Figure B.3: Isolated perilune evolution in the $e - \omega$ polar plot for representatives that were not merged in Step 5.

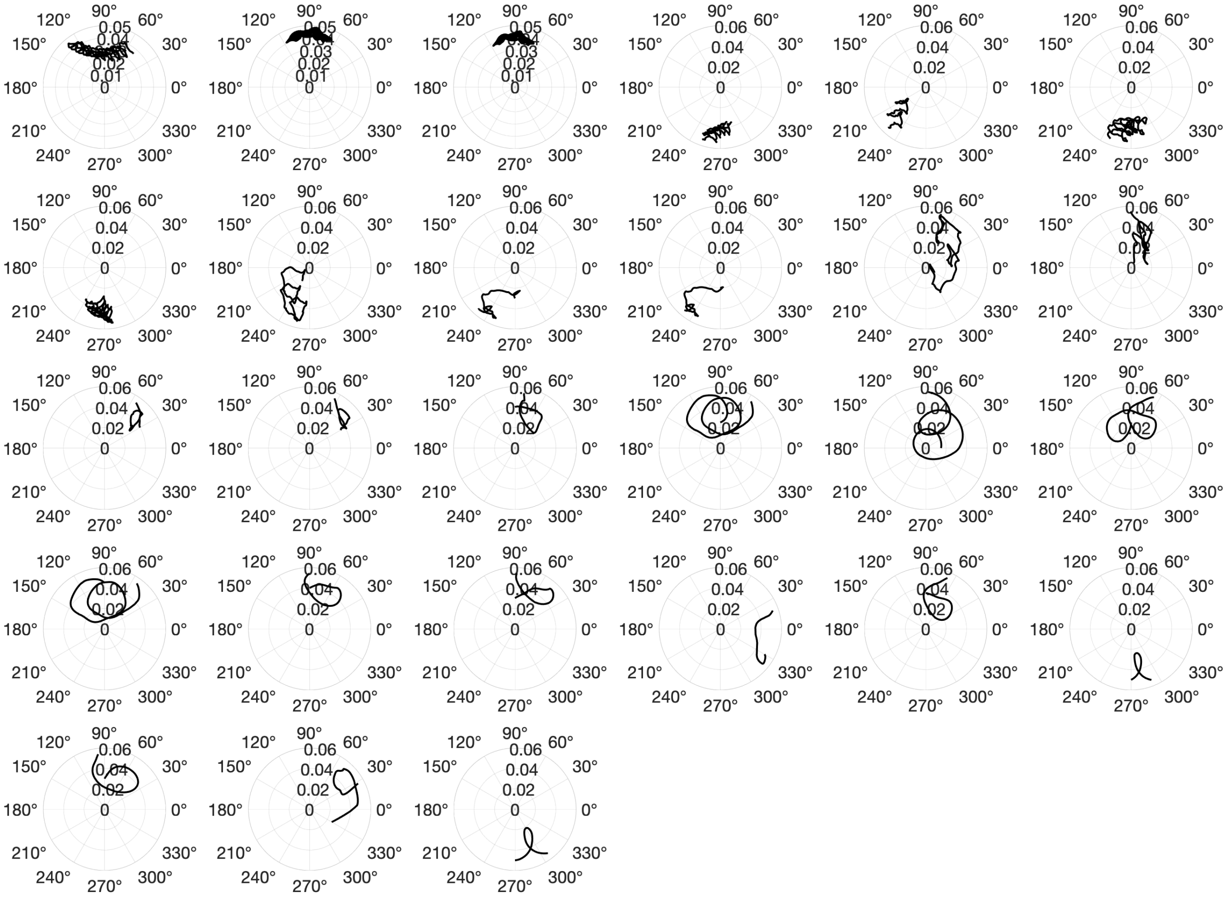


Figure B.4: Isolated perilune evolution in the $e-\omega$ polar plot for representatives that were not merged in Step 5.